

SIMULATION OF A TRACK-WHILE-SCAN RADAR

WILLIAM FREDERICK DELANEY

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

~~DO NOT DOWNGRADING~~
~~APPROVED FOR PUBLIC RELEASE~~

SIMULATION OF A TRACK-WHILE-SCAN RADAR

by

William Frederick Delaney
Lieutenant, United States Navy
B.A., Harvard University, 1958

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1967

ABSTRACT

A model of a TWS radar is developed that provides a realistic computer simulation for comparing various radar tracking methods.

Prediction accuracy of a simplified alpha - beta tracker is compared to that of an adaptive filter. In addition, the effect on radar tracking of a variable gate size correlation technique is investigated.

TABLE OF CONTENTS

Chapter	Page
1. Introduction	9
2. Radar Model	11
3. Filters	20
4. Conclusions	30
Bibliography	32
Appendix	
1. Graphical output	33
2. List of variables	50
3. Logic flow diagrams	57
4. Program Listing	78

LIST OF TABLES

Table	Page
1. Search beam positions vs. known targets.	15
2. Probability of detection vs. range.	17

LIST OF ILLUSTRATIONS

Figure	Page
1. Attack profile	11
2. Simulated antenna motion - Mode 1	14
3. Beam scanning sequence for two targets engaged - Mode 2	16
4. Block diagram of filter equations	23
5. Flow graph of $1/S^2$ plant	24
6. Change in radar azimuth due to a turning target	28

CHAPTER I

INTRODUCTION

A need exists for more extensive use of computer simulations as an aid for defining and obtaining the solutions to systems engineering problems. To gain some insight and experience in this field a computer simulation of a hypothetical point defense radar system is carried out.

The criteria for defining the radar parameters and method of operation were derived from the following situation.

A destroyer class ship should be able to provide itself with adequate protection against a multiple aircraft raid while transiting singly. In order to provide timely detection and adequate intelligence for launch of a defensive weapon the shipboard radar must be capable of the following functions:

1. Full hemispherical coverage.
2. Medium range search i.e. 50-100 n.m.
3. Multiple target track while scan.
4. High data rates in critical threat sectors.

In addition, to minimize costs the radar design should be compatible with existing weapons. To provide a realistic approach, only those designs and techniques capable of being implemented with currently available components were allowed.

A phased array antenna with a monopulse beam was chosen to meet the requirement of high data rates on multiple targets. Phased arrays are inherently limited in scan. To meet the requirement of full 360° coverage it was necessary to rotate the antenna mechanically in azimuth. Formulation of the problem in this way led to the selection of two modes of operation. The first is characterized by complete

radar coverage with mechanical rotation in azimuth and electronic scanning in elevation, with monopulse in elevation angle only. This mode has limited tracking capability due to a relatively low revisit rate. Mode 2, a sector scan with an electronically steered beam in azimuth and elevation and monopulsing in two coordinates, is capable of providing 0° to 90° elevation coverage and $\pm 45^\circ$ in relative azimuth.

It is felt that the two-mode solution is justified since one can assume with some confidence that the raid will be radially inbound with little or no intentional maneuvering at ranges greater than twenty-five nautical miles. Once the raid has been detected and evaluated as such by the observer, Mode 2 can be selected to provide a high data rate to track the target, whose probability of maneuver increases as the range to the target decreases and as his weapon launch point is approached.

The switching range is not critical and can be assumed to be a function of the maximum range of the expected weapon available to the enemy aircraft (of the order of 5 to 15 n.m.) plus some safety time.

Once the radar was modeled and the computer program written, the simulation became a convenient vehicle for investigation of the radar tracking equations which are a critical facet of a TWS system. A comparison of prediction accuracy and resolution of target tracks between a classical alpha/beta tracker and an adaptive technique was pursued as an extension of the basic premise of utilizing the computer as a device for design and analysis.

CHAPTER II

RADAR MODEL

A discussion of some assumptions which affected the choice of the type of radar to be modeled was presented in the introduction. This chapter presents some of the reasoning and methods used to select basic parameters for the radar necessary to establish a reasonable time scale for the simulation. This initial section is followed by a discussion of a few of the more important subroutines in enough detail to supplement the flow diagrams of appendix II where necessary.

Choice of Basic Parameters.

Assume:

- (1) Target velocity = 600 n.m./hr.
- (2) Attack profile shown in fig.
- (3) Target weapon release point \leq 10 m.m. max.
- (4) Defensive weapon = missile (mach 3).
- (5) Missile requires a high data rate for command guidance.
- (6) Antenna height (h_a) = 50 ft.

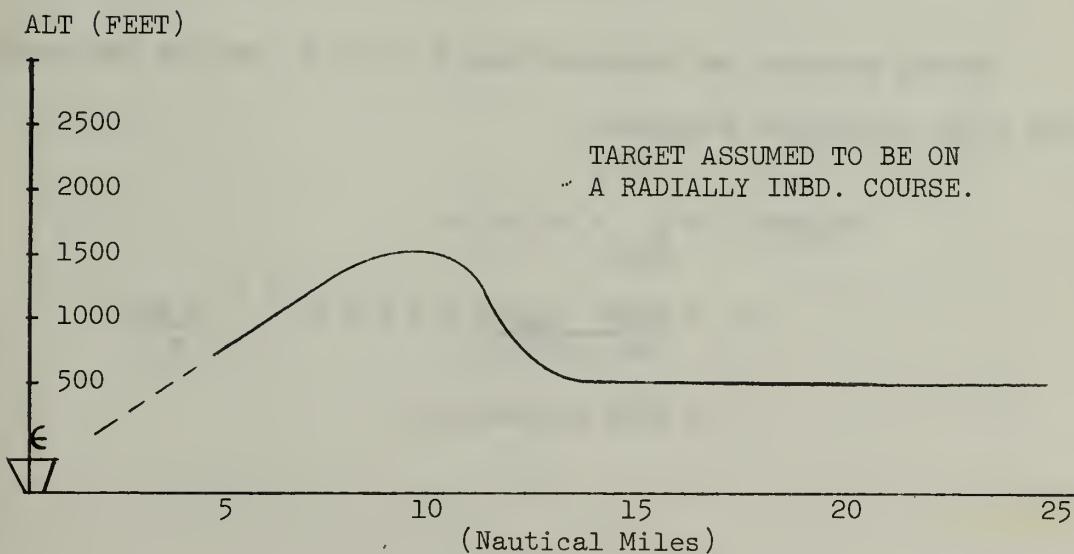


Fig. 1 ATTACK PROFILE

Desiring a 30 second sector scan in which to fire and control the defensive missle, we must commence sector scan at a minimum range of 20 n. m. This ensures an impact point of approximately 15 n. m., providing adequate time for a second attempt before the attacking aircraft reaches his weapon launch point. Arbitrarily assuming twice this range (40 n. m.) will provide adequate time for threat evaluation. Based on the radar horizon given by the following equation, 50 n. m. was chosen as the range at which a probability of detection, equal to 90% on a single scan, would provide sufficient initial detection capability.

$$r = \sqrt{2h_a} + \sqrt{2h_t} \text{ (miles)}$$

h_t = height of target A/C.

given : $h_a = 50$ ft.

$r \approx 41$ miles $h_t = 500$ ft.

$r \approx 54$ miles $h_t = 1000$ ft.

Having obtained an idealized $R_{max} = 45$ n. m., we can calculate the pulse repetition frequency.

$$R(\text{unamb.}) = \frac{c}{2f_r} = 45 \text{ n. m.}$$

$$f_r = \frac{3 \times 10^8 \text{ m/sec}}{2 \times 45 \text{ n.m.}} \times 5.4 \times 10^{-4} \frac{\text{n.m.}}{\text{m.}}$$

$$f_r = 1800 \text{ cycles/sec.}$$

Assuming a fan shaped antenna beam (1° azimuth by 5.5° in elevation) and requiring the equivalent of integrating 4 pulses per scan we determine the time for one revolution of the antenna in mode 1.

$$\frac{1}{1800 \text{ cycles/sec.}} \times \frac{4 \text{ cycles}}{1^\circ \text{ Beam}} \times \frac{360^\circ}{\text{Position REV.}} = 0.8 \frac{\text{SEC.}}{\text{REV.}}$$

This figure must be multiplied by the number of 5.5° sections in elevation that must be covered. Taking into account the target profile and realizing the advantages of short revisit rates a compromise value of 54.5° coverage in elevation requiring 8 sec. total scan time was selected.

To simulate the continuous motion of the antenna, azimuth is divided into 0.02° increments. The following diagram describes the antenna motion.

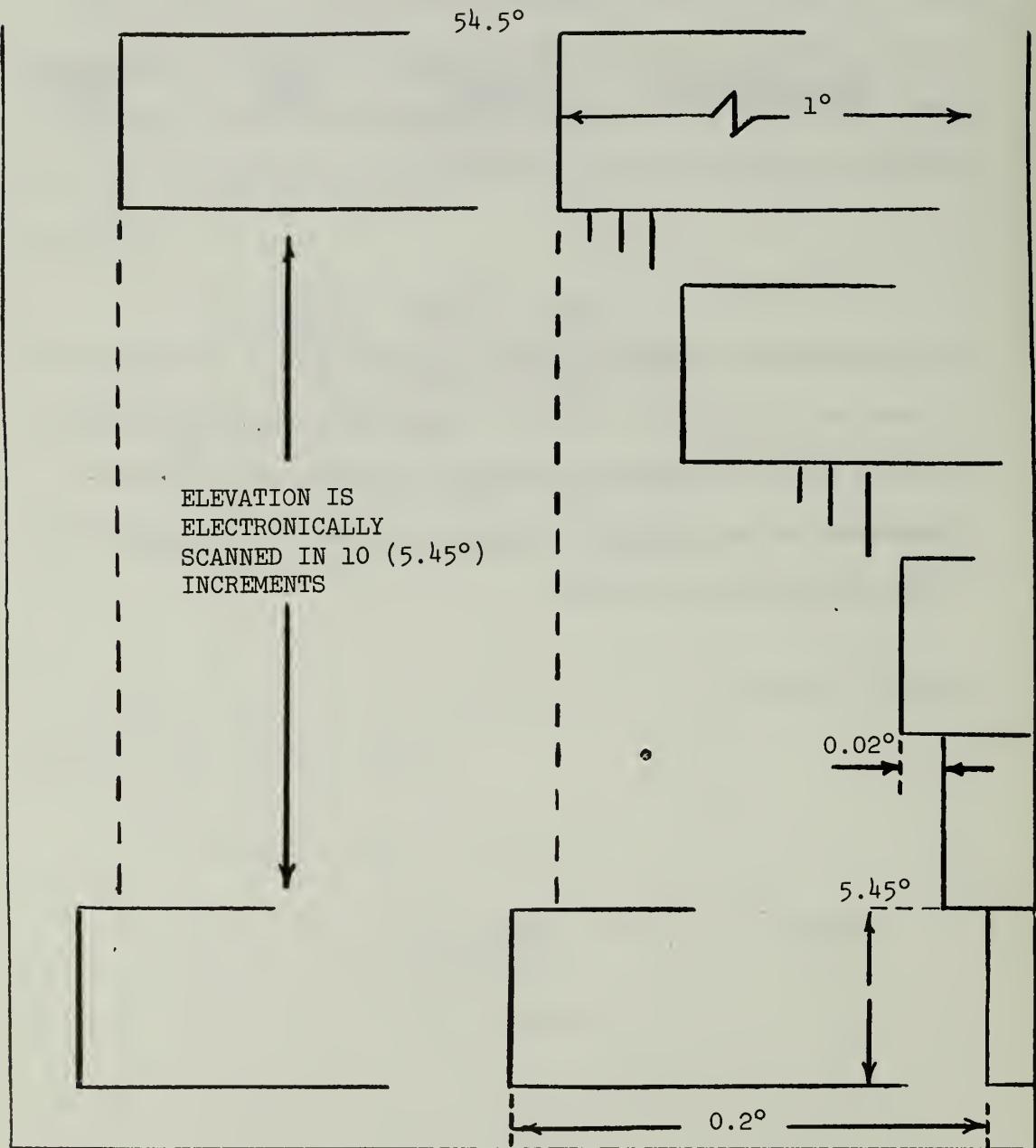


Fig. 2. SIMULATED ANTENNA MOTION — MODE 1

Turning now to a discussion of mode 2 or full electronic sector scan the following assumptions were made:

(1) Defensive missile has a 20 millisecond command guidance requirement repeated at intervals of 1 second.

(2) Special purpose beam steering computer capable of:

- (a) Calculating 200 beam positions per second.
- (b) superimposing; collimation commands on the beam position to obtain variable beam shape.
- (c) Beam shape = $2^\circ \times 2^\circ$ pencil beam.

(3) 5 ms. illumination time per beam position except 25 ms. illumination time for beam positions where targets are predicted to be present.

(4) 10 - target capability, maximum.

The above assumptions lead to the establishment of the following table:

TABLE 1. Search Beam Positions vs. known targets

Targets	Search Beam Positions Remaining Per 1 Second Interval.
0	200
2	190
4	180
6	170
8	160
10	150

In a $\pm 45^\circ$ AZ. by 60° EL. sector we have $45 \times 30 \approx 1350$ ($2^\circ \times 2^\circ$) beam positions. For the worst case of 10 targets engaged the revisit rate to a "same" search position would be $\frac{1350}{150} =$ approximately 9 seconds. This scan interval is reduced to 6.75 seconds for the case of 0 targets engaged.

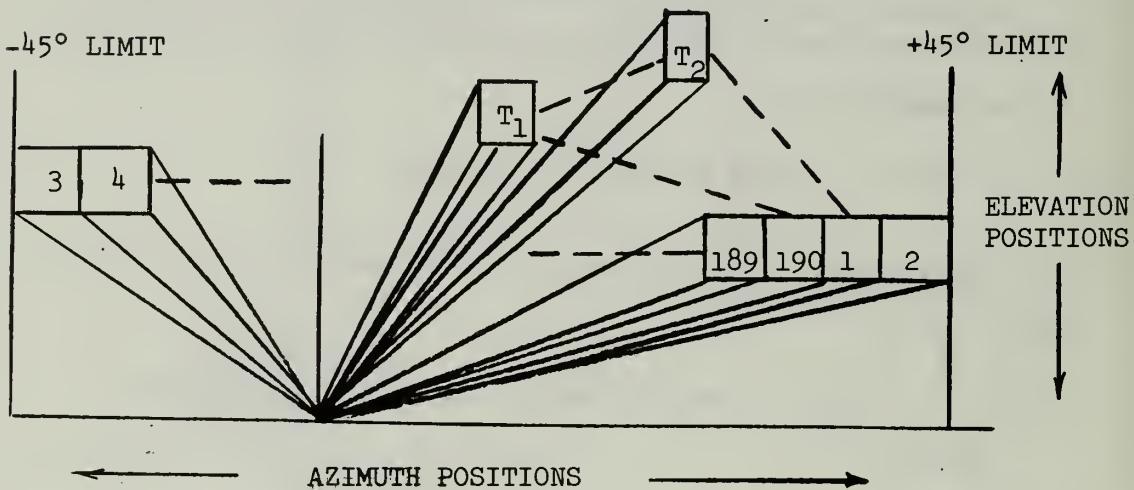


Fig. 3. BEAM SCANNING SEQUENCE FOR 2 TARGETS ENGAGED. MODE 2.

The target detection model for mode I and mode II are essentially identical and can be described in the following way. If the radar beam passes over a target the target range as given by the target generator is compared to the values in Table 2 to determine the probability of detection for that range. If Random, the result of calling the uniform random number generator RAN1, is less than the probability of detection, we assume a detection; otherwise a miss or noise return.

TABLE 2. Probability of detection vs. range.

Prob. of Det.	Target Range
0.22	60.0
0.5	55.0
0.6	54.0
0.7	53.0
0.8	52.0
0.85	51.0
0.9	50.0
0.95	49.0
0.99	48.0
0.999	47.0

Having established a radar target the radar range RADRNG is computed by the equation RADRNG = TARGRNG + RANERR where RANERR is obtained by calling the GAUSSIAN random number generator and using the equation:

$$\text{RANERR} = \text{DEV} * 0.3$$

We can see from the above equations that the radar range is equal to the true range plus an error term with a standard deviation by choice, of 0.3 n. m.

It should be noted at this point that all radar measurements are computed in essentially the same manner. That is RADRNG, AZRAD and ELRAD are computed as noisy observations of the "true" target position, as given by the target generation subroutine, TARGEN.

CORRELATION:

To correlate current radar observations with tracks established from predictions on previous observations, subroutine CORRASS was devised. Current radar observations are stored in the matrices RADRNG, AZRAD, ELRAD, While predicted values from the tracker are stored similiarly. Once during each revolution of the antenna each track is compared with each observation; for each observation that a track agrees with, a one is placed in the corresponding location of the C(I,J) correlation matrix. The columns of the correlation matrix correspond to tracks and the rows correspond to observations. For a track to agree with an observation certain tolerances were assumed: range must be within $\pm X$ n.m., azimuth and elevation angles must be within $\pm Y$ degrees.

X and Y are determined as follows:

(1) Alpha - Beta filter.

Range gate = 2.5 n.m.

Azimuth gate = 2.5 n.m.

Elevation gate = 1.5 n.m.

(2) Kalman Filter.

Range gate = $2.0 * \text{SQRTF} [P(1,1)]$ n.m.

Azimuth gate = $1.0 * \text{SQRTF} [P(3,3)]$ n.m.

Elevation gate = $2.5 * \text{SQRTF} [P(5,5)]$ n.m.

After this initial correlation several observations may correlate with one track or several tracks may correlate with one observation. In these situations, a set of four association rules are applied to

the correlation matrix, $C (I,J)$, to assign observations to the proper tracks.

These rules implemented by subroutine ASSOC, are as follows:

- (1) A track which correlates with several observations rejects any observation held in common with another track, if the common observation is the only observation correlating with the second track.
- (2) A track correlating with several observations, some of which are not held in common with other tracks, rejects observations which are held in common with other tracks.
- (3) When several observations correlate with one track, the closest observation is associated with that track.
- (4) When several tracks correlate with one observation, the observation is associated with the closest track.

Rules 3 and 4 are based on the value of range since it has the smallest $\sqrt{2}$ for measurement noise. The above correlation and association routine is essentially the same as that given by [7].

CHAPTER III

FILTERS

Smoothing of raw radar reports and prediction of future observations, basic functions of an automatic tracking system, can be performed conveniently by a set of equations implemented on a computer. In this simulation two distinct schemes were chosen to perform these functions. In the simplest case target tracks are based on smoothing and prediction of an alpha-beta tracker operating in a cartesian coordinate reference frame.

The smoothing equation is:

$$\hat{\Lambda}_{sn} = \hat{\Lambda}_{pn} + \alpha (\tilde{\Lambda}_n - \hat{\Lambda}_{pn})$$

The prediction equations:

$$\hat{\Lambda}_n = \hat{\Lambda}_{n-1} + \frac{\beta}{\tau} (\tilde{\Lambda}_n - \hat{\Lambda}_{pn})$$

$$\hat{\Lambda}_{pn+1} = \hat{\Lambda}_{sn} + \hat{\Lambda}_n * T$$

Where: $\hat{\Lambda}_{sn}$ = Smooth value of X,Y or Z for the n^{th} scan.

$\hat{\Lambda}_{pn}$ = Predicted value of X,Y or Z for the n^{th} scan.

$\tilde{\Lambda}_n$ = Noisy observation of X,Y or Z for the n^{th} scan.

$\hat{\Lambda}_n$ = Smooth prediction of X,Y or Z component of velocity for the n^{th} scan.

Where: Λ_{pn+1} = Predicted X, Y or Z coordinate for the $N + 1$ scan.
T = Time between looks which is essentially constant.
T = 1 second for sector scan.
T = 8 seconds for full scan.
 α = The smoothing parameter.
 $\beta = \frac{\alpha^2}{2-\alpha}$ for optimum filtering.

Normally associated with the alpha-beta smoothing prediction equations are a set of rules for determining bin size. The rules are usually based on track firmness, length of time since last correlation and radar range. Basically this set of rules for varying the correlation gate size is an attempt to make the system adaptive to target dynamics.

Experimentation with the simple alpha-beta tracker indicated that given a reasonable fixed gate size based only on accuracy of the radar measurements, tracks could be maintained on targets with mild maneuvers, ie., less than $3^\circ/\text{sec.}$ turns, with little difficulty given a suitable value of alpha. An important result observed from these simulations was that the real problem associated with tracking is the time delay before sensing a target's maneuver initiation.

Investigation of a criterion for determining target maneuvers seemed to be the next logical step. Changes in the magnitude and angle of the target velocity vector in the X - Y plane were looked at briefly as one possibility. From the data accumulated, incidental to other tasks being performed, the changes in angle appeared to be an unlikely candidate due to extreme noise caused by measurement errors of the radar. The magnitude of the vector, on the other hand, appeared to be rather insensitive to target maneuvers.

In lieu of pursuing this somewhat uncertain investigation, it was decided to use techniques that were in comparison well defined. The Kalman filter was chosen as the best solution to the problem for the following reasons:

(1) While the Kalman filter can not predict target maneuvers, the P matrix generated by the filter equations is directly influenced by target maneuverability (Q matrix) and at the same time is a measure of our confidence in predicted values. It seemed reasonable, therefore, to make the correlation gate size proportional to the corresponding values of the P matrix.

(2) If an observation is missed due to a target maneuver we can expand the gate size to some larger value to increase the probability of correlation on the next scan with automatic gate size reduction if we succeed.

The Kalman Filter Equations as given in (4):

$$G_n = P_{n/n-1} H^T [H P_{n/n-1} H^T + R]^{-1}$$

$$\hat{X}_{n/n} = \hat{X}_{n/n-1} + G_n [z_n - H \hat{X}_{n/n-1}]$$

$$P_{n/n} = P_{n/n-1} - G_n H P_{n/n-1}$$

$$\hat{X}_{n+1/n} = \hat{X}_{n/n}$$

$$P_{n+1/n} = \emptyset P_{n/n} \emptyset^T + Q$$

These equations represented in block diagram form appear in fig. 4.

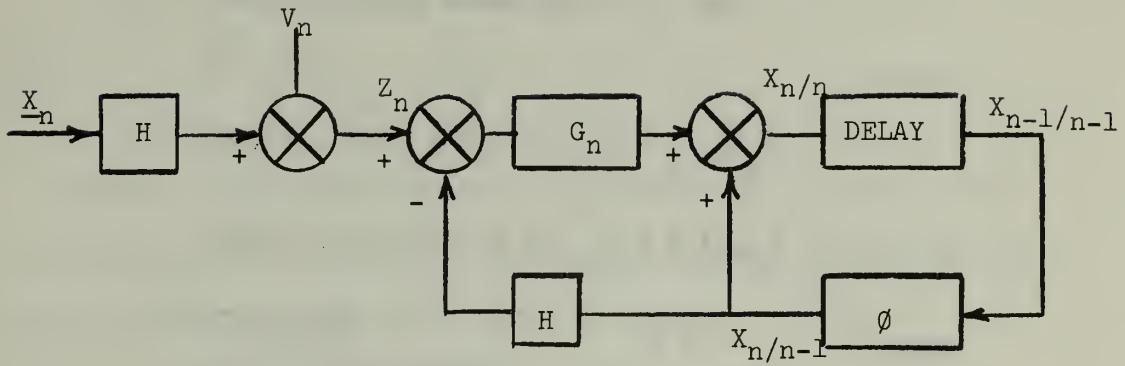


Fig. 4. BLOCK DIAGRAM OF FILTER EQUATIONS.

where: $Z = H X + V$

or $Z = \begin{bmatrix} \text{Radar Range} \\ \text{Radar Azimuth} \\ \text{Radar Elevation} \end{bmatrix}$

V represents the noise associated with the radar measurement of the three observables and is gaussian.

H = observability matrix.

or $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

\emptyset = The state transition matrix that represents the dynamics of the aircraft target. Assuming the aircraft can be represented as pure inertia of a $\frac{1}{s^2}$ plant.

We have

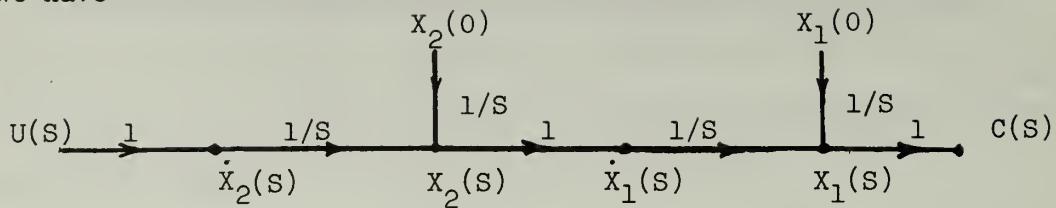


Fig. 5. FLOW GRAPH OF $1/s^2$ PLANT.

Yields:

$$\dot{x}_1(s) = 0 x_1(s) + 1 x_2(s) + 0 U(s)$$

$$\dot{x}_2(s) = 0 x_1(s) + 0 x_2(s) + 1 U(s)$$

hence:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow [SI-A] = \begin{bmatrix} s & -1 \\ 0 & 1/s \end{bmatrix}$$

where: $\Delta_A = \frac{1}{s^2}$

$$[SI-A]^{-1} = \frac{\begin{bmatrix} s & 0 \\ -1 & s \end{bmatrix}^T}{\Delta} = \frac{\begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix}}{\Delta} = \begin{bmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{bmatrix}$$

we know: $\phi(s) = [SI-A]^{-1} = \begin{bmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{bmatrix}$

hence: $\phi(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$

Assuming the components of the X vector are uncoupled.

$$\phi(t) = \begin{bmatrix} 1 & t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

R = The covariance of the measurement noise is by definition = $E [V V^T]$

In this case the standard deviation of the measurement

noise on Range, Azimuth and elevation are known to have the following values.

Range = 0.3n.m.

Az = 0.5 deg.

El = 0.5 deg.

hence:

$$R = \begin{bmatrix} 0.09 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

$\underline{X}_{n/n-1}$ represents the predicted states of a radar target.

To initialize the filter for a particular target, that is specify $\underline{X}_{1/0}$, the following method is used.

$$\underline{X}_{1/0} = \begin{bmatrix} \text{Radar Range} \\ \dot{R} = -0.166 \\ \text{Radar Azimuth} \\ \dot{AZ} = 0.0 \\ \text{Radar Elevation} \\ \dot{El} = 0.0 \end{bmatrix}$$

This method assumes the target is closing the radar radially ($\dot{AZ} = 0.0$) at a low altitude ($\dot{El} \approx 0.0$) with a velocity of approximately 600 n.m. /hr.

$P_{n/n-1}$ represents the amount of uncertainty in the predicted values of target states $X_{n/n-1}$.

$$P = E [(\underline{X}_n - \hat{\underline{X}}_{n/n-1})(\underline{X}_n - \hat{\underline{X}}_{n/n-1})^T]$$

where: \underline{X} is the true state.

$\hat{\underline{X}}$ is the predicted value.

To initialize the filter we must provide.

$P_{1/0}$ Given $\underline{X}_{1/0}$ defined previously and R the covariance of the measurement noise;

$P(1,1)$, $P(3,3)$, $P(5,5)$ are known.

Again relying on the assumption of a radially closing target at low

altitude with a speed of 600 m.m./hr. we can assume the remaining diagonal elements of P which represent $\frac{V_R^2}{R}$, $\frac{V_{AZ}^2}{AZ}$, $\frac{V_{EL}^2}{EL}$ will be small. Actual values were determined empirically and as a result the initial covariance of error matrix is determined.

$$P = \begin{bmatrix} 0.09 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0278 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00109 \end{bmatrix}$$

It should be noted here that $P_{1/0}(3,3) = 2.5$ is used in place of 0.25 to aid tracking maneuvering targets. If an observation fails to correlate with an established track the P matrix for that track is set equal to the above values. The gate sizes, and in particular, the Azimuth gate, are directly related to the P matrix values. This procedure enlarges the gate size thus increasing the probability of correlation on the next scan.

Lastly but certainly not of the least importance is the Q matrix specification. The Q matrix which represents perturbations to the state vector due to target maneuvers must be representative of the full range of movements the target is expected to use.

Values for Q were estimated by assuming a "worst" case such as 6000 ft./min. rate of climb, 1 g. linear acceleration and 12° / second turning rates. One half these "worst" case values were actually used in an attempt to average the Q matrix over the full range of expected maneuvers.

Estimation of $Q(1,1)$

Assume the aircraft can vary speed by (100 n.m / hr. max.)

$\cong \pm 0.027$ n.m./sec.

$Q \triangleq E [\underline{W} \underline{W}^T]$ where \underline{W} is a random signal caused by target maneuvers.

Taking $1/2 (0.027)$ implies

$$Q(1,1) = (0.013) (0.013) = 0.0169$$

Estimation of $Q(2,2)$

Assume the aircraft can accelerate at ($1g$) max.

Taking $1/2 (1g) \cong .0027 \frac{\text{n.m.}}{\text{sec}^2} \times 1 \text{ sec.}$

implies that $Q(2,2) = (.0027) (.0027) \cong .00001$

Estimation of $Q(3,3)$

Assume the aircraft can make a (12° / second) max. turn.

Taking $1/2 (12^\circ/\text{sec}) = 6^\circ/\text{sec.}$ and using fig. 6

We have:

$$\text{chord length} = 2r \sin \theta/2$$

$$\text{or } \theta = 2 \sin^{-1} \frac{.02}{30} \cong 0.083^\circ$$

Hence.

$$Q(3,3) = (.083) (.083) = .00692$$

Estimates for the remaining terms of Q were obtained in a like manner with Q taking the following form.

$$Q \text{ est.} = \begin{bmatrix} 0.0169 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.00001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0069 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0011 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0011 \end{bmatrix}$$

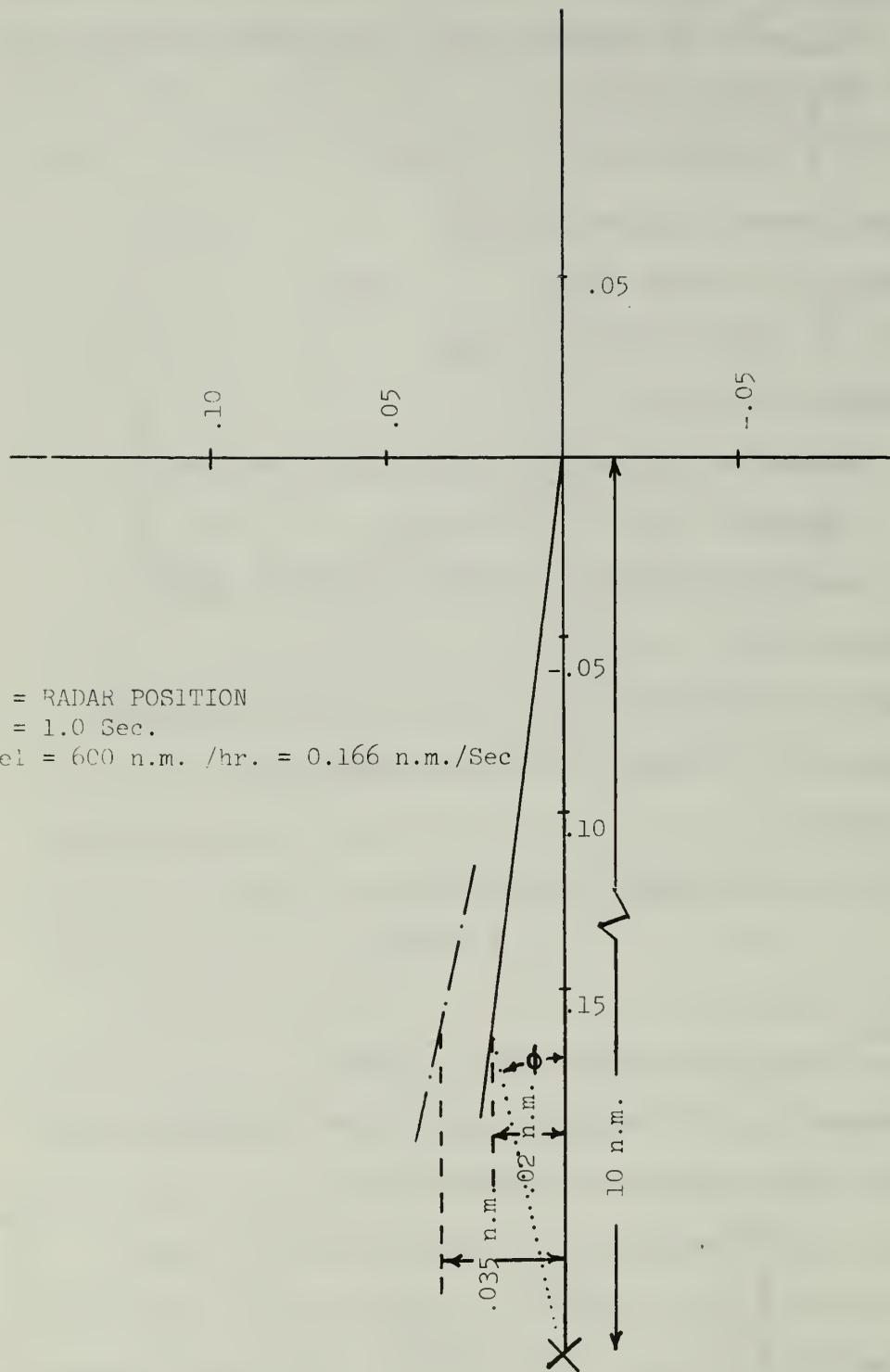


Fig. 6. CHANGE IN RADAR AZIMUTH DUE TO A TURNING TARGET.

Simulations were run to optimize the Q values in order to obtain the best filter response over the range of 0° , 3° , 6° , and 12° /sec. turns with constant speed targets.

$$Q \text{ exp.} = \begin{bmatrix} 0.00169 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0030 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.015 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.00001 \end{bmatrix}$$

The Q_8 matrix values where not increased in magnitude as might at first be expected but were set equal to Q_1 for the following reasons:

(1) The eight second scan time applies to targets of fairly long range compared to target ranges anticipated for sector scan, and as such the targets are expected to maneuver as little as possible in an attempt to close the ship.

(2) It is intuitively felt that $T(\text{ave.})$ for aircraft maneuvers for this type of target might be expected to be 3-5 seconds vice 8 seconds.

(3) The effect of maneuvers acting on a spherical coordinate system decreases with increasing range.

CHAPTER IV

CONCLUSIONS

The radar system simulation presented in this paper is felt overall to be a simplified but realistic model of a sophisticated system which could be implemented with current "state of the art" hardware.

The basic main program / subroutine organization divides the system into easily recognized functions capable of being extended in scope, modified in part, or replaced entirely with a new concept, while maintaining the integrity of the remainder of the program. Any portion of the model can be reworked to meet special requirements of a particular problem of interest.

By the addition of new subroutines which might include phenomena associated with clutter, target scintillations, and atmospheric attenuation a more realistic detection probability, given a set of initial conditions, could be achieved.

Hence the original objective of building a model for simulation of a hypothetical radar system as a tool for analysis of a variety of radar system problems has been met.

As an exercise with practical significance, the simulation was used to investigate the relative effectiveness of several types of radar tracking filters.

While no pretension of an extensive analysis is implied, the author feels that a reasonable and unequivocal comparison of the filters can be made from the material presented.

Based on the ensemble averages of the squared difference between predicted and actual positions the Kalman filter obviously provides a somewhat better tracking response for all target tracks tested.

The tracking ability of both filters appears to be about equal for "look alike" targets in close proximity. The ability of the tracking routine to resolve these targets depends on using relevant gate sizes. The gate size in turn is directly related to the accuracy of the radar measurement of range, and angles, and to the extent of maneuver capability of the target. The minimum gate size allowable without inducing excessive non-correlation due to observation noise would thus be essentially the same for both filters.

The most significant advantage of Kalman Filter tracking appears to be in its ability to track maneuvering targets without dropping track. This advantage is a result of its ability to automatically increase the gate size if a firm track fails to correlate, thus improving the probability of correlation on the next scan.

The requirement for peripheral tracking functions such as track firmness and quality would surely be reduced in a system using Kalman filtering since the (P) matrix can be used as a measure of this type of information. This feature affects somewhat the disadvantage of increased computation time due to the recursive algebraic equations required to calculate the G and P matrices.

Finally, the dependence of the Kalman filter on the Q matrix dictates that an efficient and reliable method for estimating Q for a particular target is essential.

BIBLIOGRAPHY

1. Povejsil, D. J. Raven, R. S. and Waterman, P. Airborne Radar, D. Van Nostrand Company, Inc. New York 1961.
2. Skolnik, M. I. Introduction to Radar Systems, McGraw-Hill Book Company Inc. New York 1962.
3. Dinneen, G. P. and Reed, I. S. An Analysis of Signal Detection and Location by Digital Methods IRE Transactions March 1956.
4. Benedict, T. R. and Bordner, G. W. Synthesis of an Optimal Set of Radar Track- While- Scan Smoothing Equations. IRE transactions on Automatic Control July 1962.
5. Lee, R. C. K. Optimal Estimation, Identification, and Control Research Monograph No. 28 The M. I. T. Press, Cambridge, Mass. 1964.
6. Briana, A. M. and Delaney, J. F. Multifunction Phased- Array Radar, P70, Raytheon Company, Bedford, Mass. 28 March 1966.
7. Titus, H. A. EE473 Class notes USNPGS Monterey, Calif. April - May 1967.

APPENDIX I

GRAPHICAL OUTPUT

Graph		Page
Alpha-beta filter	0°/sec turn rate.	36
Kalman filter	0°/sec turn rate.	37
Alpha-beta filter	3°/sec turn rate.	38
Kalman filter	3°/sec turn rate.	39
Alpha-beta filter	6°/sec turn rate.	40
Kalman filter	6°/sec turn rate.	41
Alpha-beta filter	12°/sec turn rate.	42
Kalman filter	12°/sec turn rate.	43
Alpha-beta filter	Crossing targets T = 8 sec.	44
Kalman filter	Crossing targets T = 8 sec.	45
Alpha-beta filter	Maneuvering target T = 8 sec.	46
Kalman filter	Maneuvering target T = 8 sec.	47
Alpha-beta filter	Crossing targets T = 1 sec.	48
Kalman filter	Crossing targets T = 1 sec.	49

Interpretation and constraints

(1) The first eight graphs provide a comparison of the prediction accuracy of each of the two filters investigated.

The following rules apply:

- (a) Four different target tracks are employed i.e. 0° , 3° , 6° and 12° /sec.
- (b) Target turning rates are initiated at $T = 6$ sec.
- (c) Each graph represents the results in range (\square), azimuth (X), and elevation (Δ) of a "Monte Carlo" ensemble average of 100 runs.
- (d) Sample rate = 1 sec. appropriate to Mode 2 operation.
- (e) Alpha-beta filtering is performed in a cartesian coordinate reference frame while Kalman filtering is done in spherical coordinates.

Filtering of the azimuth coordinate, which happens to be changing most rapidly, is about equal for both filters. The Kalman filter is obviously superior in range and elevation predictions.

The following symbol table applies for all succeeding graphs.

_____	true track
X	predicted position target one.
Δ	predicted position target two.

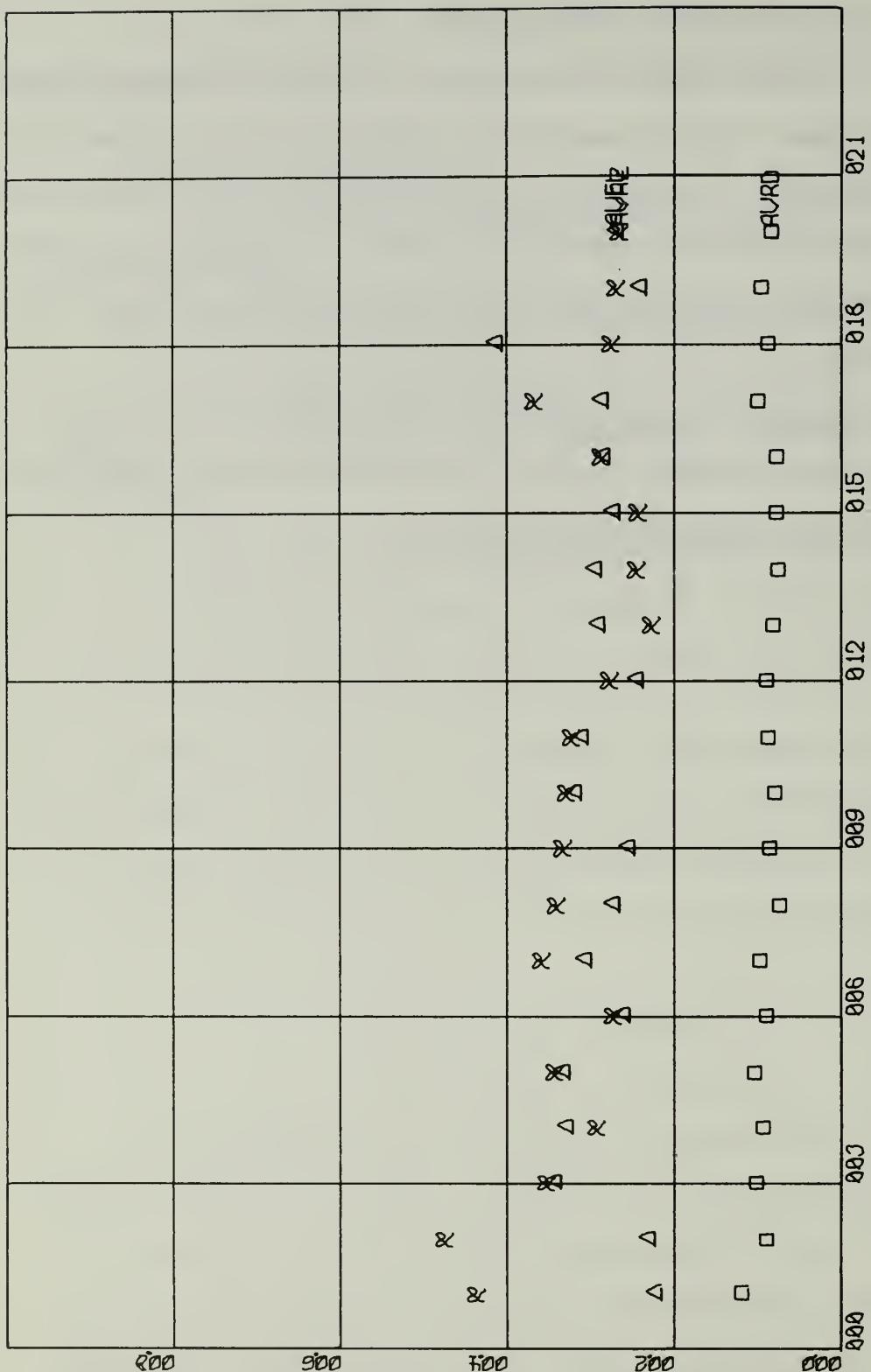
(2) Graphs nine and ten represent filtering ability for "look alike" targets. i.e. similar range, elevation, and azimuth.

The ability of both filters, supplemented by a set of correlation and association rules, to distinguish between similar targets in close proximity leaves something to be desired. A reduced revisit rate i.e. (1 sec vice 8 sec.) has little or no effect. (see graphs 13 and 14).

It seems reasonable therefor to assume that a higher resolution radar is the only solution to this problem.

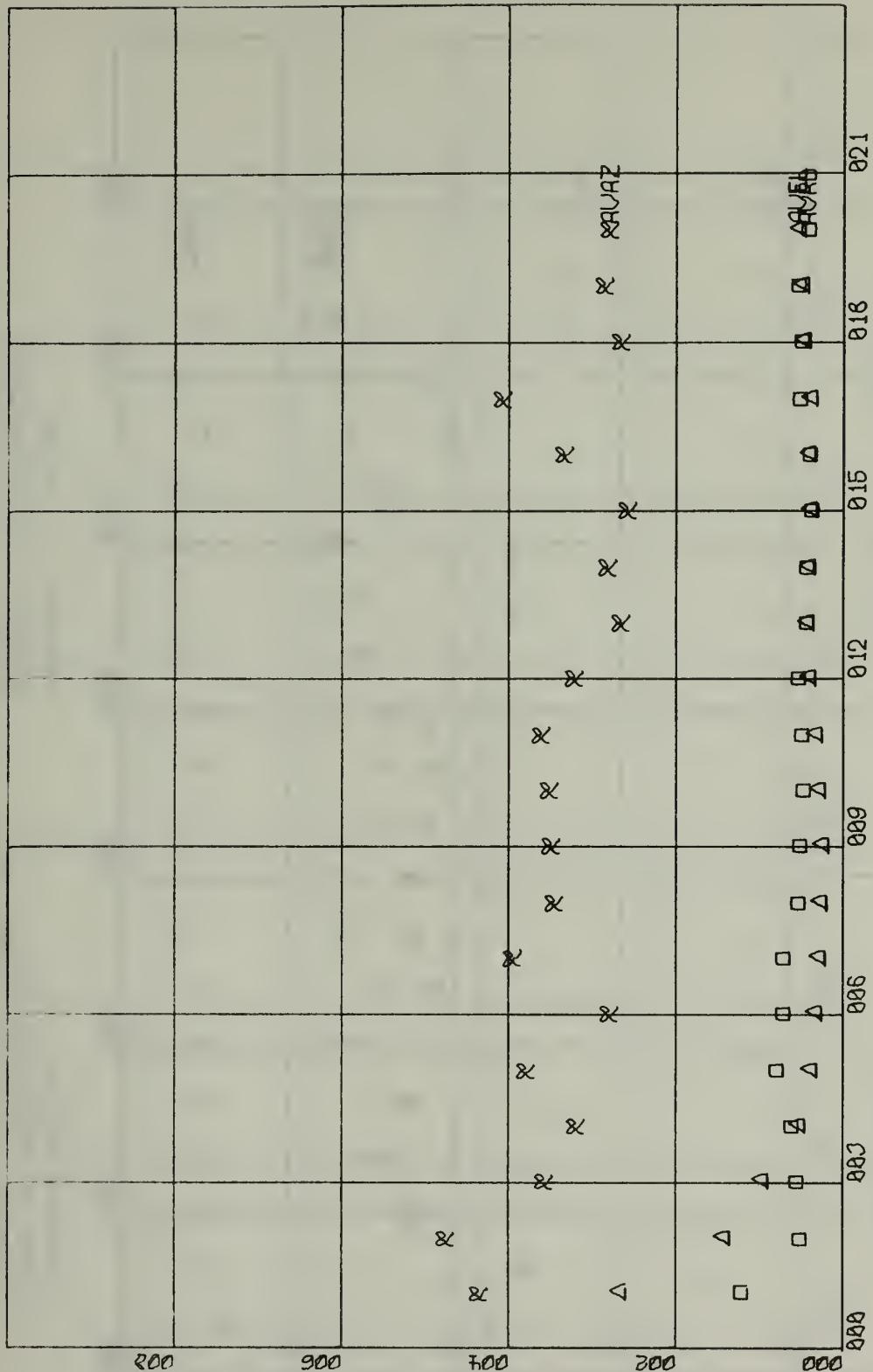
(3) Graphs ten and eleven indicate the improved tracking response of the Kalman filter for maneuvering targets, aided by a variable gate size technique. Investigation of the printed output for this run shows that the Kalman filter operating on spherical coordinates consistently predicted the position of the turning target within the limits of the gate size.

In contrast the Alpha-beta filter was unable to follow azimuth changes with sufficient accuracy to obtain the necessary correlations and as a result dropped track excessively.



ALPHA-BETA FILTER
 X-TIME IN INTEGER SECONDS
 Y-AVERAGE ((PREDICTED - ACTUAL)**2)

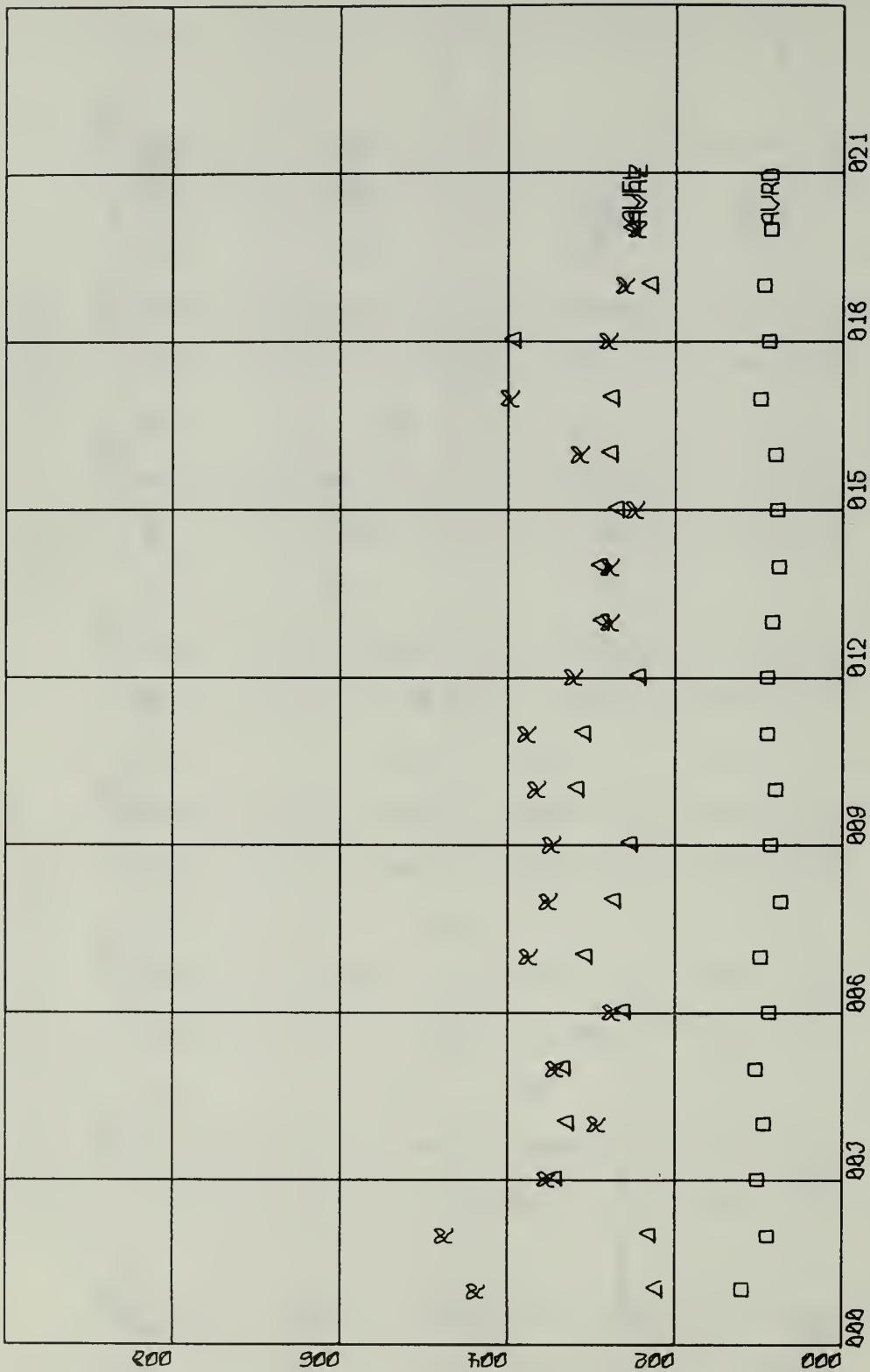
TURN RATE 0 DEG/SEC.
 X-SCALE - 3.00 UNITS/INCH.
 Y-SCALE - 0.20 UNITS/INCH.



KALMAN FILTER

X - TIME IN INTEGER SECONDS
 Y - AVG.((PREDICTED - ACTUAL)**2)

TURN RATE 0 DEG/SEC.
 X-SCALE - 3.00 UNITS/INCH.
 Y-SCALE - 0.20 UNITS/INCH.

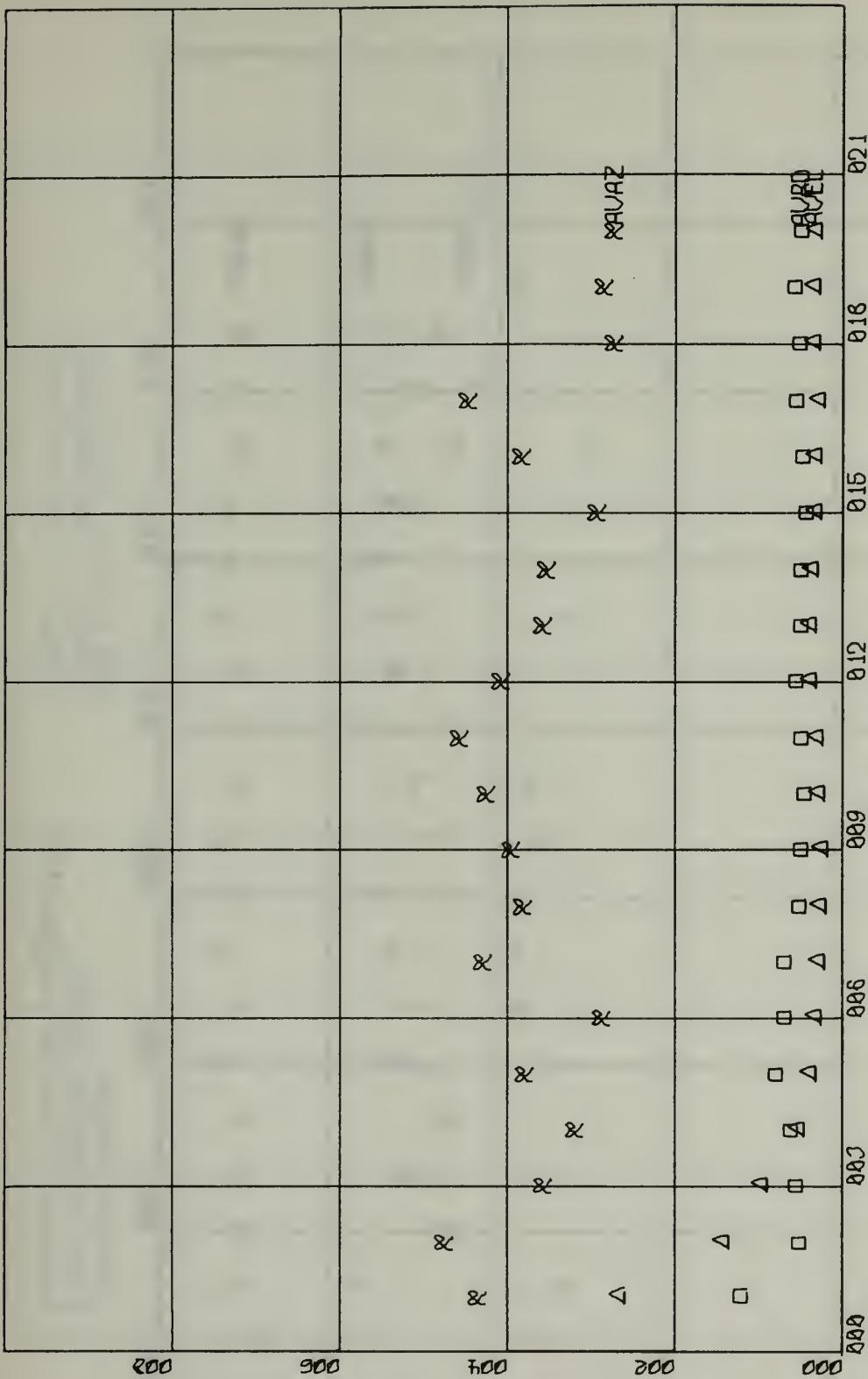


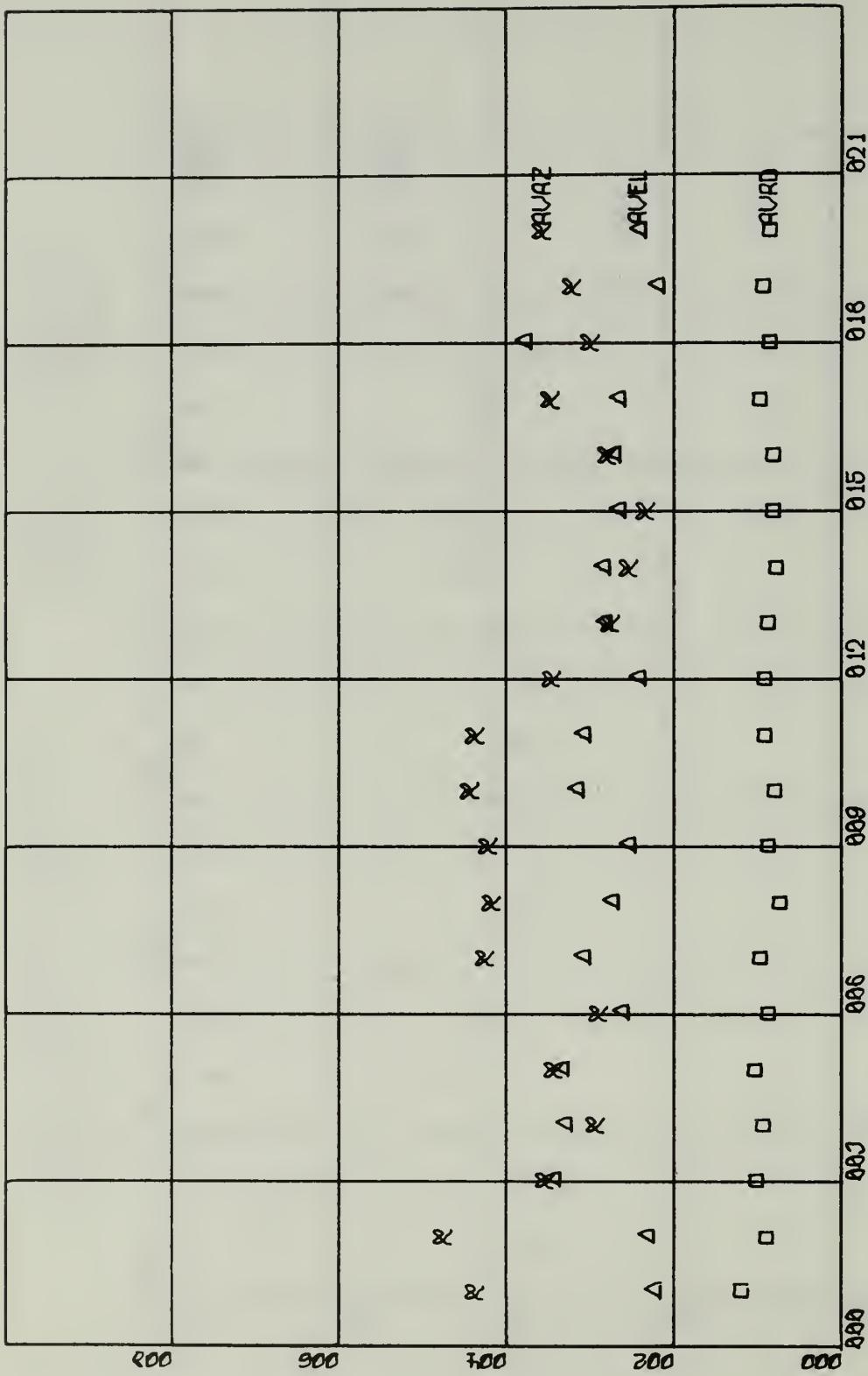
ALPHA-BETA FILTER
 X-TIME IN INTEGER SECONDS
 Y-AVERAGE (PREDICTED - ACTUAL) * * 2)

TURN RATE 3 DEG/SEC.

X-SCALE - 3.00 UNITS/INCH.

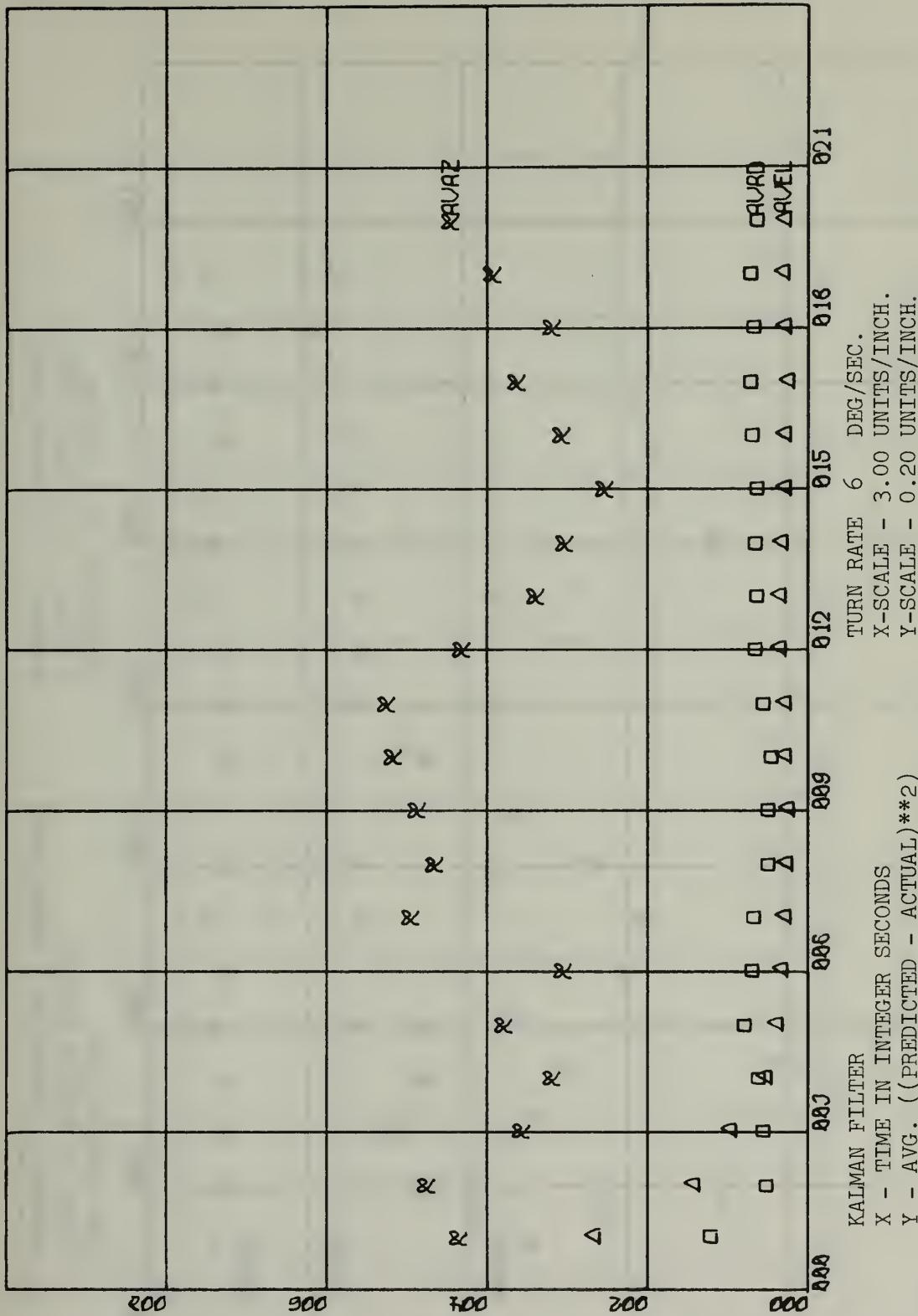
Y-SCALE - 0.20 UNITS/INCH.

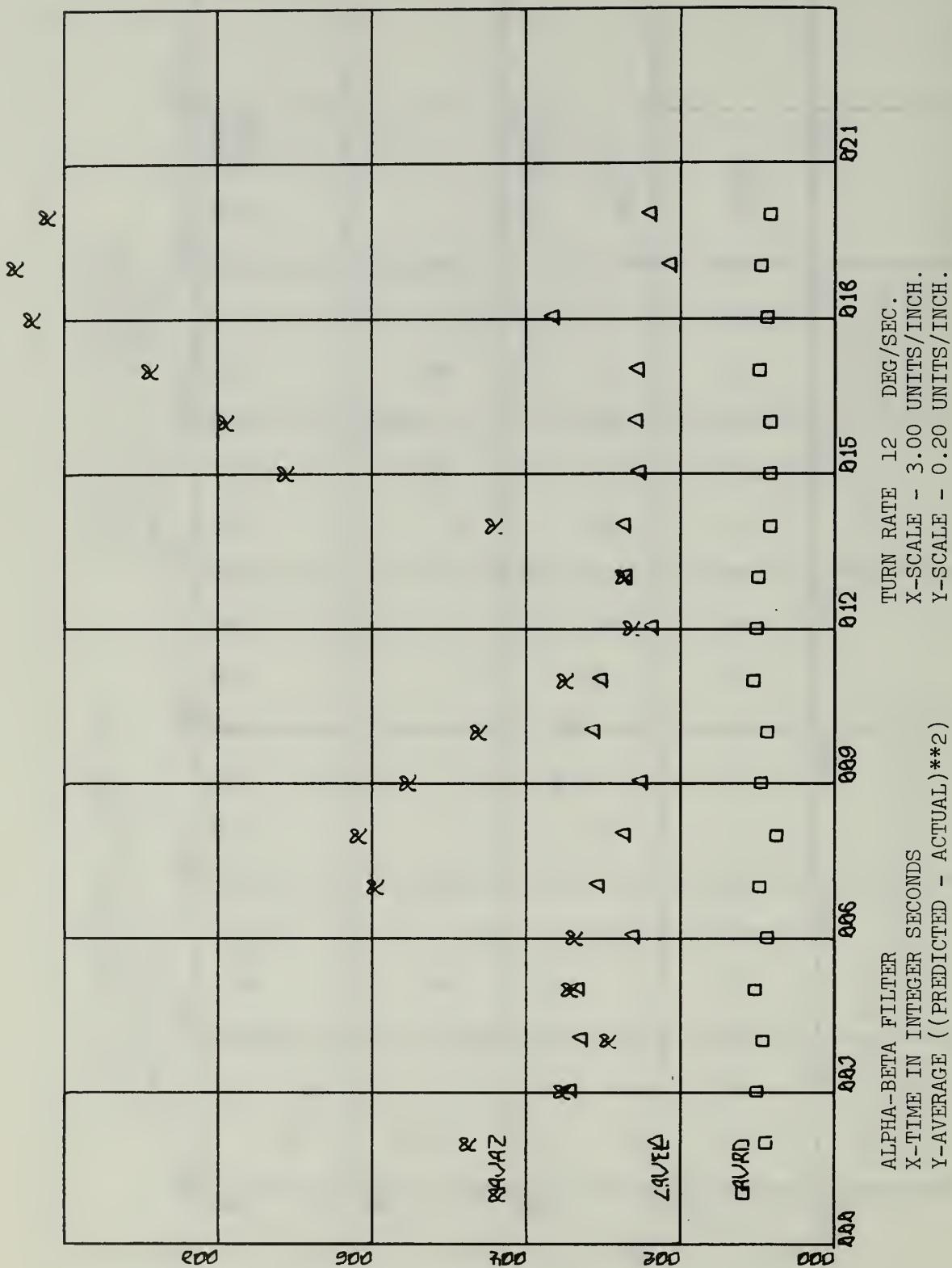


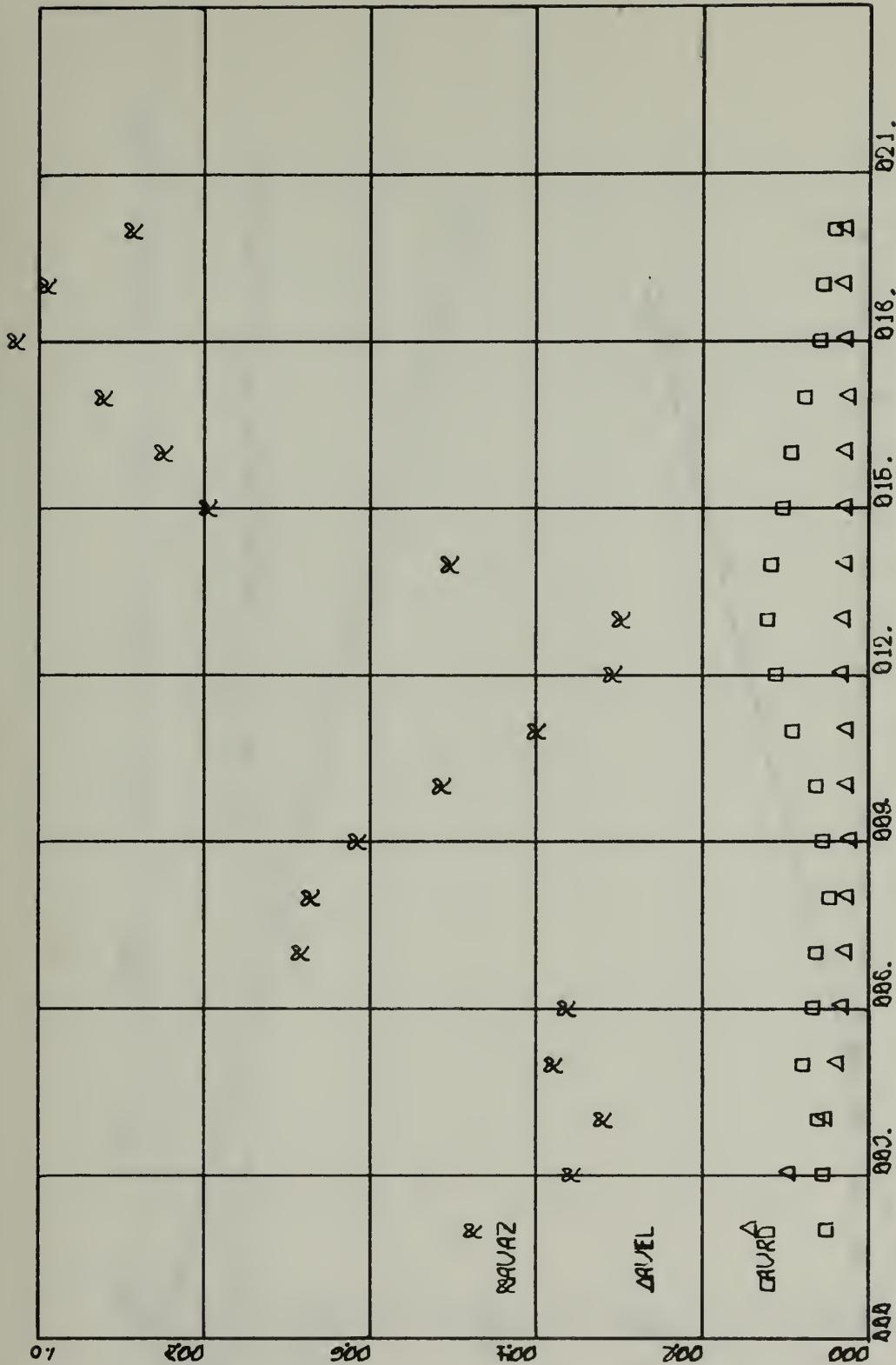


ALPHA-BETA FILTER
X-TIME IN INTEGER SECONDS
Y-AVERAGE ((PREDICTED - ACTUAL)**2)

TURN RATE 6 DEG/SEC.
 X-SCALE - 3.00 UNITS/INCH.
 Y-SCALE - 0.20 UNITS/INCH.

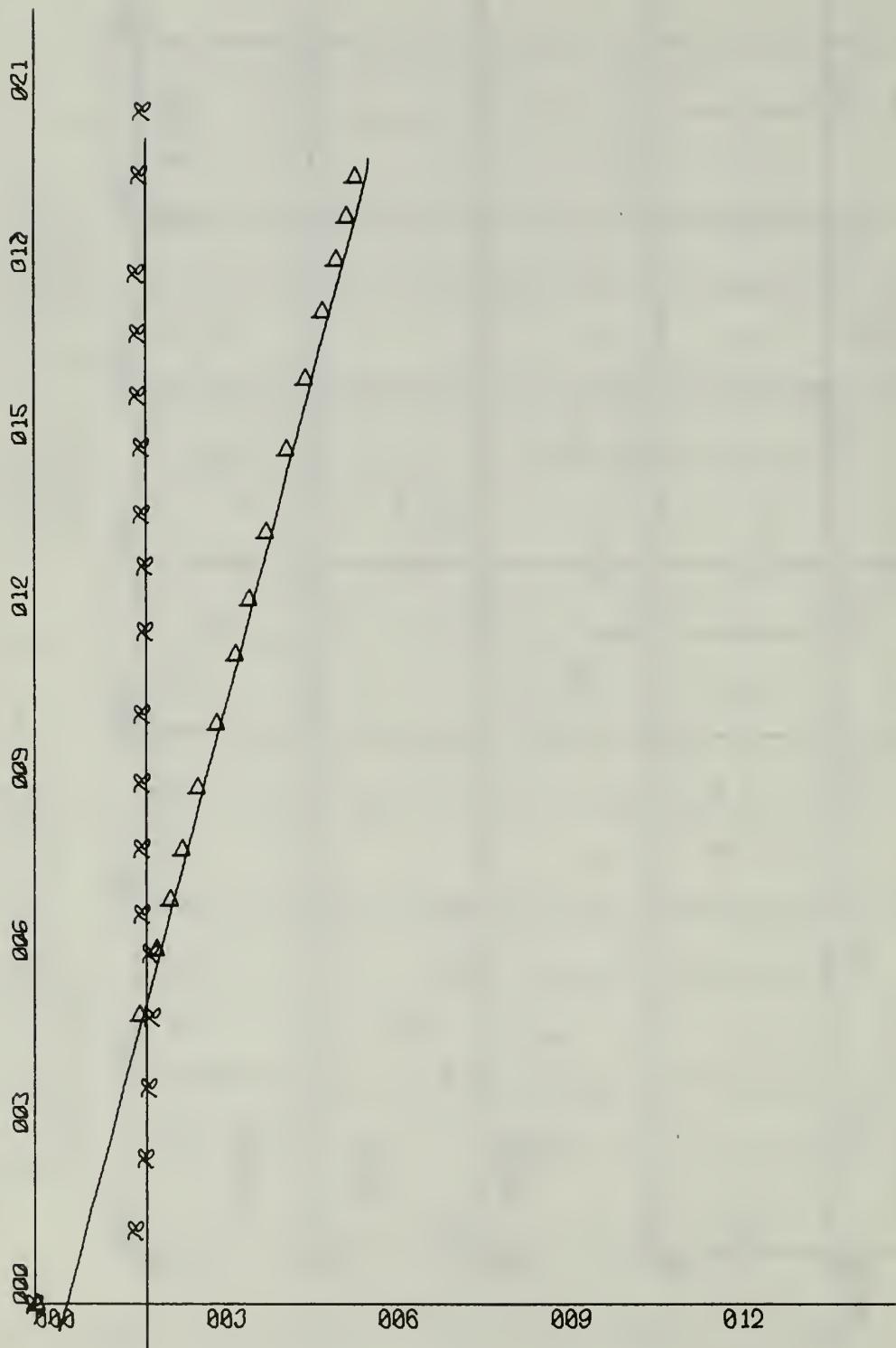


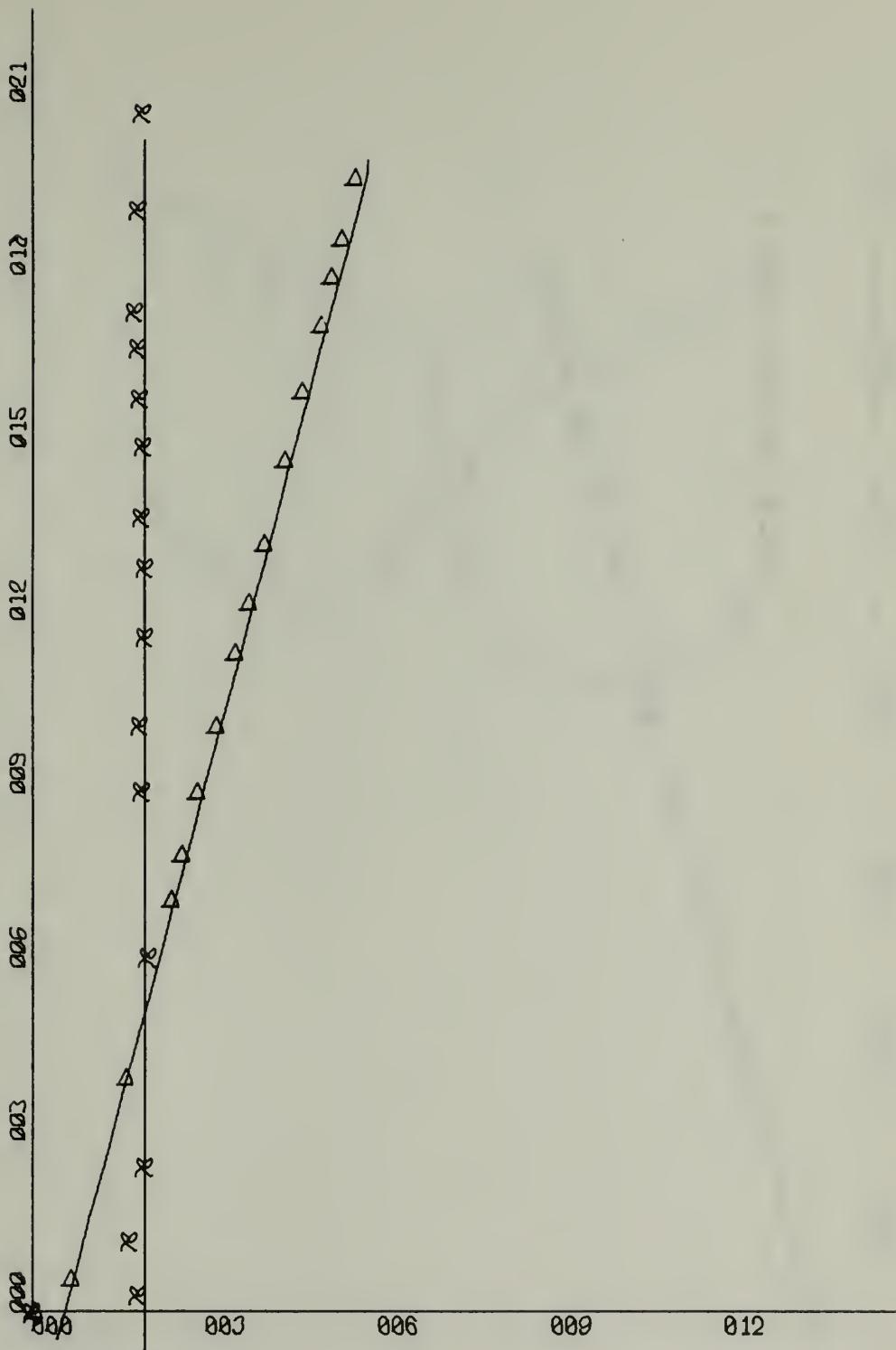




KALMAN FILTER
 X - TIME IN INTEGER SECONDS
 Y - AVG. ((PREDICTED - ACTUAL)**2)

TURN RATE 12 DEG/SEC.
 X-SCALE - 3.00 UNITS/INCH.
 Y-SCALE - 0.20 UNITS/INCH.





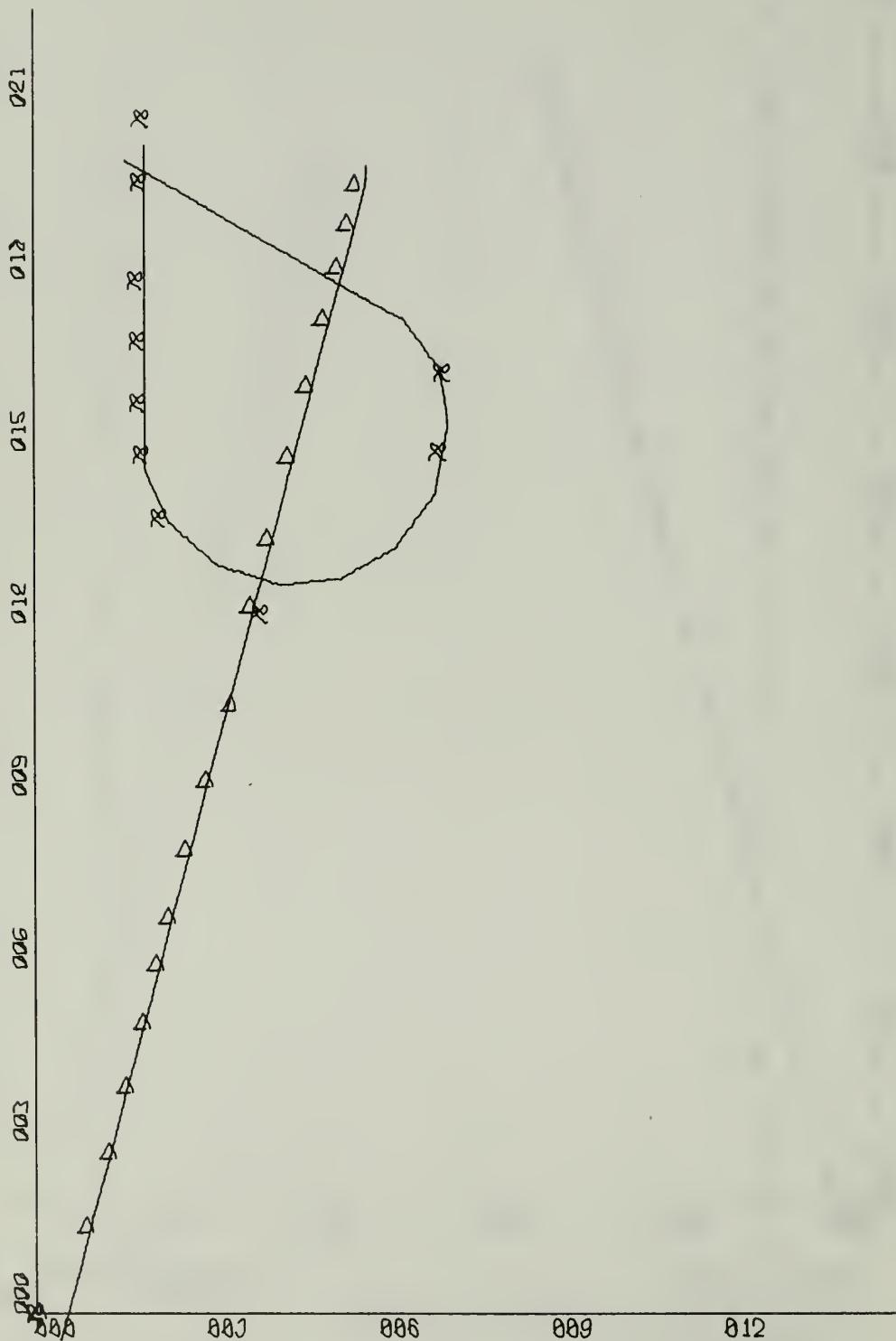
KALMAN FILTER

X = Y = N.M.

X SCALE = Y SCALE = 3.00 UNITS/INCH

NON MANEUVERING

T = 8 SEC.

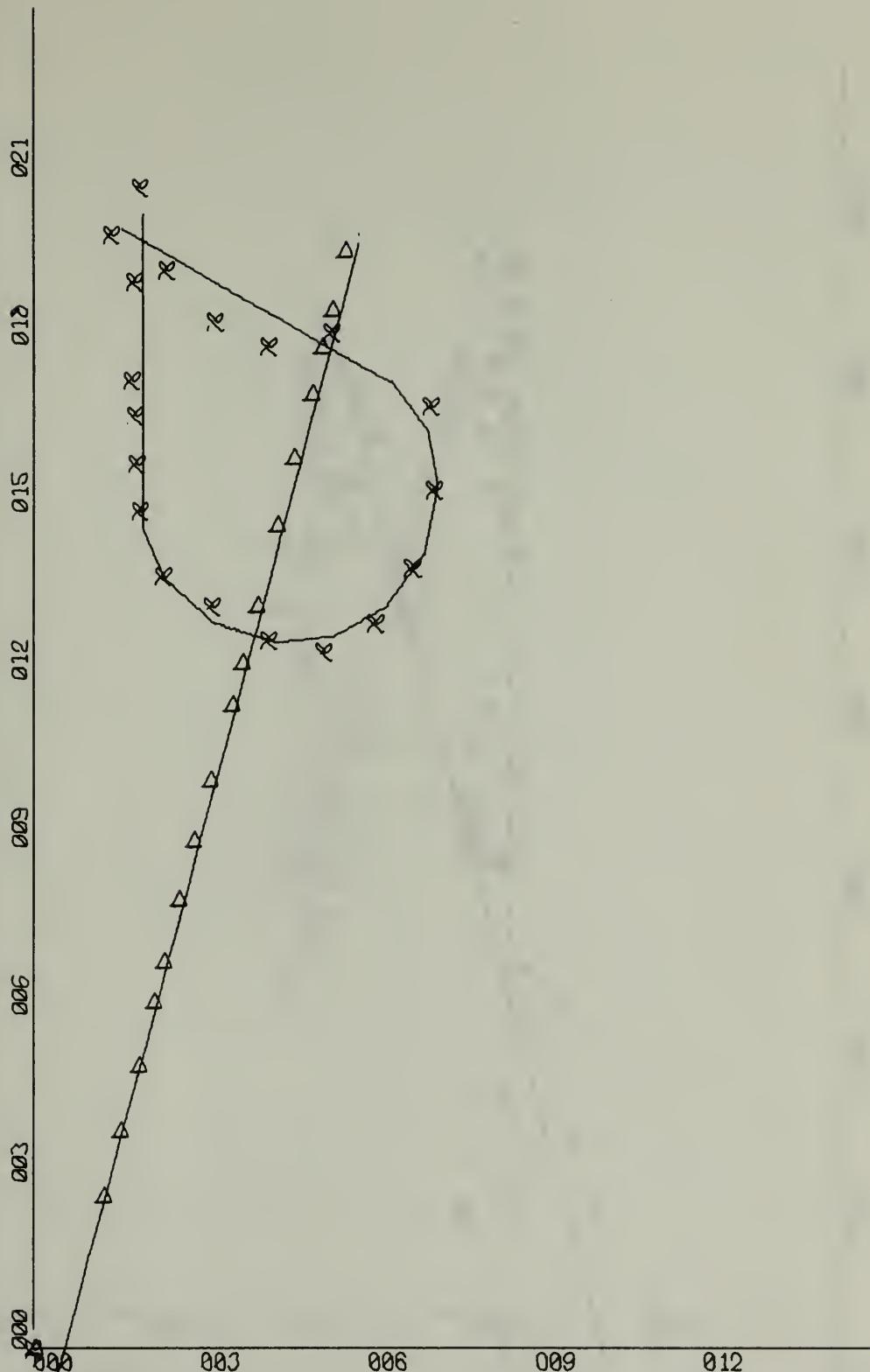


ALPHA-BETA FILTER

X = Y = N.M.

X SCALE = Y SCALE = 3.00 UNITS/INCH.

3 DEG/SEC



KALMAN FILTER

X = Y = N.M.

X SCALE = Y SCALE = 3.00 UNITS/INCH.

3 DEG/SEC



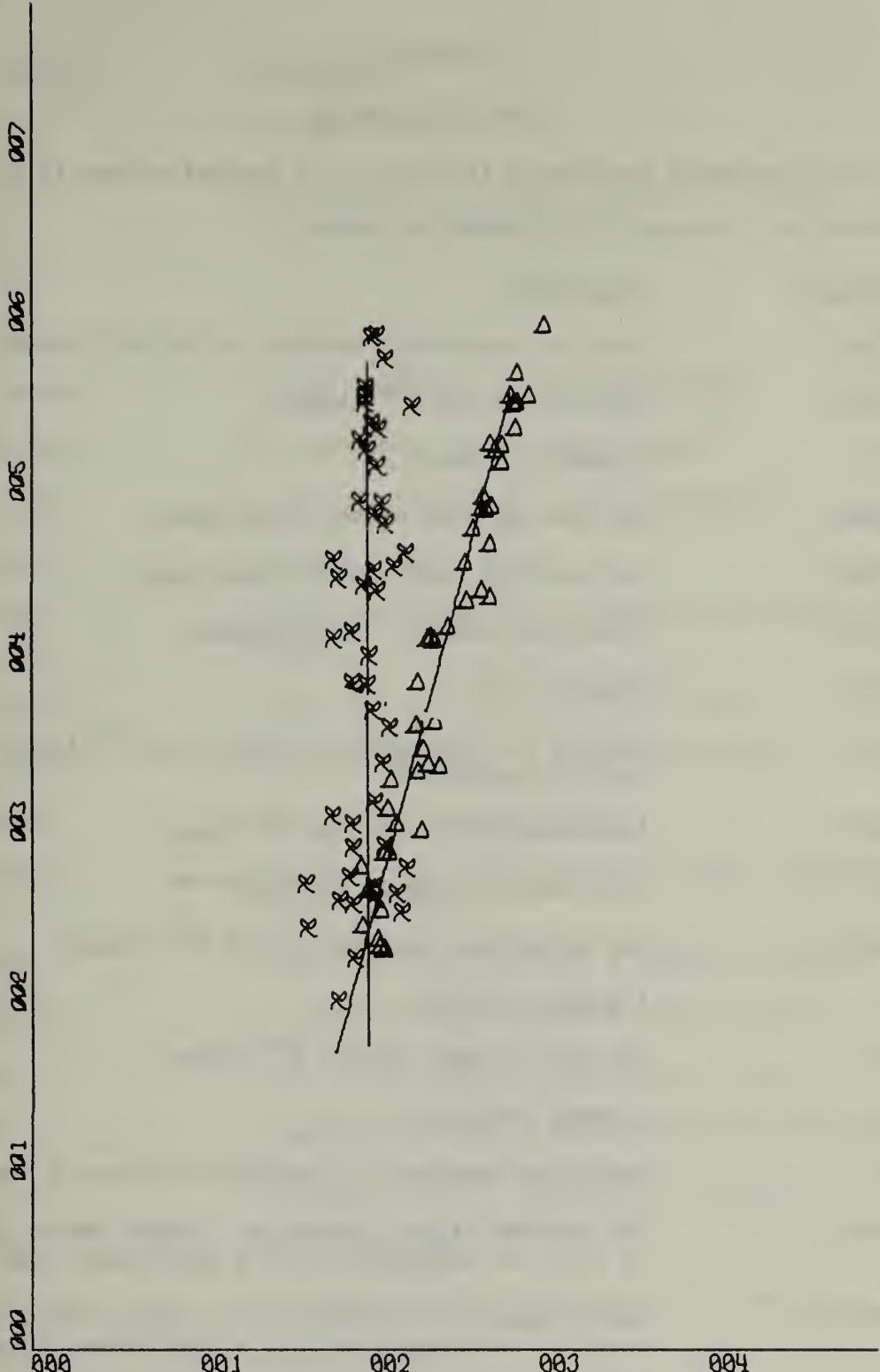
ALPHA-BETA FILTER

X = Y = N.M.

X SCALE = Y SCALE = 1.00 UNITS/INCH

NON MANEUVERING

T = 1 SEC.



KALMAN FILTER

X = Y = N.M.

X SCALE = Y SCALE = 1.00 UNITS/INCH

NON MANEUVERING

T = 1 SEC.

APPENDIX II

LIST OF VARIABLES

This appendix contains a listing of the variables used in the simulations, arranged in alphabetical order.

<u>Variable</u>	<u>Definition</u>
ALPHA	Position smoothing constant of the α -B tracker.
ALT(K)	Altitude of the K^{th} target.
ALT1	A dummy variable.
AZBEAML	The leading edge of the radar beam.
AZBEAMT	The trailing edge of the radar beam.
AZDIFF	Predicted azimuth - true azimuth.
AZDOT	Azimuth rate.
AZ4(K)	Counter for the number of times the K^{th} track fails to correlate.
AZG(K)	Azimuth gate size of the K^{th} track.
AZ2(I)	The ensemble average of ELDIFF **2
AZRAD(K)	The noisy radar azimuth of the K^{th} target.
AZI	A dummy variable.
AZT(K)	The true azimuth of the K^{th} target.
AZY	A dummy variable.
BETA	Smoothing constant for velocity in the α -B tracker
BOXMAX (I)	The maximum value allowed for a radar return if it is to be associated with a particular range box.
BOXMIN(I)	The minimum value allowed for a radar return if it is to be associated with a particular range box.
C(I,J)	Observation / track correlation matrix.
CONT(K)	Azimuth of the leading edge of the radar beam on initial detection of the K^{th} target.
DELTA	Normalized velocity vector.

<u>Variable</u>	<u>Definition</u>
DOTT	A dummy variable.
DOTX	The predicted velocity components of the K^{th} target as given by the OX -B tracker.
DOTY	.
DOTZ	.
ELBEAMB	Bottom extemity of the radar beam.
ELBEAMT	Top extemity of the radar beam.
ELDIFF	Predicted elevation - true elevation.
ELDOT	Elevation angle rate.
ELERR	Radar measurement noise added to the true elevation angle.
ELG(K)	Elevation gate size of the K^{th} track.
ELRAD(K)	Noisy radar value of target elevation.
ELT(K)	True target elevation angle.
ELTEST	Parameter for distinguishing between range boxes by elevation angle.
FLAG	Flag is set by MOVAVE if a target is detected.
G(I,J)	Gain matrix for the Kalman filter equations.
GO	Used by SCAN 360 to compress full scan.
H(I,J)	Observability matrix for Kalman filter equations.
HD(K)	Heading of the K^{th} aircraft in degrees.
HPHT	A dummy variable.
HT(I,J)	The transpose of the observability matrix.
IA	Range box index.
IAT(I)	Used as a counter in conjunction with IAZ.
IAZ(I)	Radar return signal counter which is compared to a threshold value to determine whether or not a detection has occured.
IAZREF	Reference azimuth of sector scan.

<u>Variable</u>	<u>Definition</u>
II	Elevation beam position indicator.
INDE(I)	SEE INDEX (I).
INDEX (I)	INDEX in conjunction with INDE determine when a track fails, two consecutive times, to correlate thus initiating drop track procedure.
ITEL	The fixed point form of TEL.
ITELI	A dummy variable.
ITEMP	The fixed point form of TEMP.
ITEMPL1	A dummy variable.
KC(I)	A flag used to initialize the Kalman filter for a particular observation.
KER	Flag to indicate an improper matrix inverse operation.
LIMB	Lower limit of the sector scan.
LIML	Port limit of the sector scan.
LIMR	Starboard limit of the sector scan.
LIMT	Upper limit of the sector scan.
NB	The number of runs.
NBT	Number of discrete times.
NO	Counter for the number of target bins currently active.
NR(I)	Flag to indicate known targets in sector scan.
NT	Number of targets.
NUMB	Sector scan counter.
NUM1	Dummy variables.
NUM2	.
NUM3	.
NUMSCAN	The number of 360 degree scans of the radar.
NUMTRK	Number of current active tracks.

<u>Variable</u>	<u>Definition</u>
OISE	A dummy variable for the noisy radar observations (Range, Azimuth or Elevation.)
P(I,J)	The covariance of error matrix.
PAZRAD	The predicted radar azimuth of the target.
PAZY	A dummy variable.
PD(I)	A discrete value of probability from the probability of detection table.
PDET	Probability of detection.
PELRAD	Predicted value of the target elevation.
PHI(I,J)	The state transition matrix used in the Kalman filter.
PHI1(I,J)	The state transition matrix for the one second case.
PHI8(I,J)	The state transition matrix for the eight second case.
PHIT(I,J)	Transpose of the PHI matrix.
PITCH(K)	Value of the nose attitude of the K th target in degrees.
PITCH1	A dummy variable.
PK1.....9(I,J)	Storage for P matrix associated with each track.
PRADRNG	Predicted radar range of a target.
PRED	A dummy variable used in the Alpha-Beta tracker for the predicted value of range, azimuth or elevation.
PXRAD	Predicted X,Y,Z, components of target position.
PYRAD	.
PZRAD	.
Q(I,J)	Matrix representing target maneuver capability.
Q1(I,J)	Q may be set equal to Q1 or Q8 as required by the appropriate scan time.
Q8(I,J)	.
R(I,J)	Covariance matrix of the observation noise.

<u>Variable</u>	<u>Definition</u>
RADDEG	Constant used to convert radians to degrees.
RADIFF	Predicted range - true range.
RANDOM	Uniformly distributed random number.
RANERR	Error in measuring target range.
RANGE(I)	Discrete value of range used to enter the probability of detection table.
RADRNG(K)	Noisy target range as given by the radar.
RADRNG1	A dummy variable.
RDOT	Range rate.
RG(K)	Range gate size for the K^{th} track of the correlation routine.
S(I)	A dummy variable used for temporary storage.
SMOOTH	The filtered radar observation in range, azimuth or elevation.
SUM	Temporary value of the radar beam leading edge.
T	Time interval for updating targets equal to scan time.
T1	Dummy variables.
T2	.
TARGRNG	True value of target range as given by the target generator.
TEL	Temporary sector scan beam reference.
TEST	.
TESTT	.
TNT	.
THETA	Value of target heading in radians with reference to the positive X-axis.
VEL(K)	Speed of the K^{th} target in n.m./hr.
VMAG2(I)	Ensemble average of radiff.

<u>Variable</u>	<u>Definition</u>
VNB	Fixed point form of NB.
VPHI2(I)	Ensemble average of azdiff.
XP(I,J)	Predicted state vector.
XPI	A dummy variable.
XRAD(K)	X coordinate of the K th target as given by the radar.
XRAD1(K)	Stored radar track quantity for graphical output.
XRAD2(K)	.
XS	Smooth state vector.
XT(K)	The true X coordinate of the K th target
XT1(K)	Stored target track quantity for graphical output.
XT2(K)	.
YRAD(K)	Radar Y coordinate of the K th target.
YRAD1(K)	Stored radar track quantity for graphical output.
YRAD2(K)	.
YT(K)	The true Y coordinate of the K th target.
YT1(K)	Stored target track quantity for graphical output.
YT2(K)	.
Z(I,J)	Measurable quantities input to the Kalman filter equations.
ZRAD(K)	The radar Z coordinate of the K th target.
ZT(K)	The true Z coordinate of the K th target.

Note: Variables associated with the following input/output and matrix algebra subroutines, have not been defined, since they are essentially dummy variables.

1. ADD
2. PRINNT
3. PROD
4. READD
5. RECIP
6. TRANS

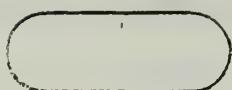
APPENDIX III
LOGIC FLOW DIAGRAMS

This Appendix contains a series of Logic Flow diagrams as listed below.

MAIN PROGRAM

SUBROUTINE	TARGEN
SUBROUTINE	SCAN 360
SUBROUTINE	DETECT
SUBROUTINE	MOVAVE
SUBROUTINE	CORRASS
SUBROUTINE	ASSOC
SUBROUTINE	ABFILT
SUBROUTINE	KALFILT
SUBROUTINE	SETUP
SUBROUTINE	TWS
SUBROUTINE	MONTE

FLOW CHART SYMBOLS



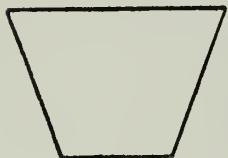
A connector or terminal.



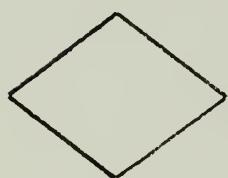
An offpage connector.



A predefined process or module/subroutine.
A more detailed flow chart of this subroutine
is also included.



Input/output other than display.

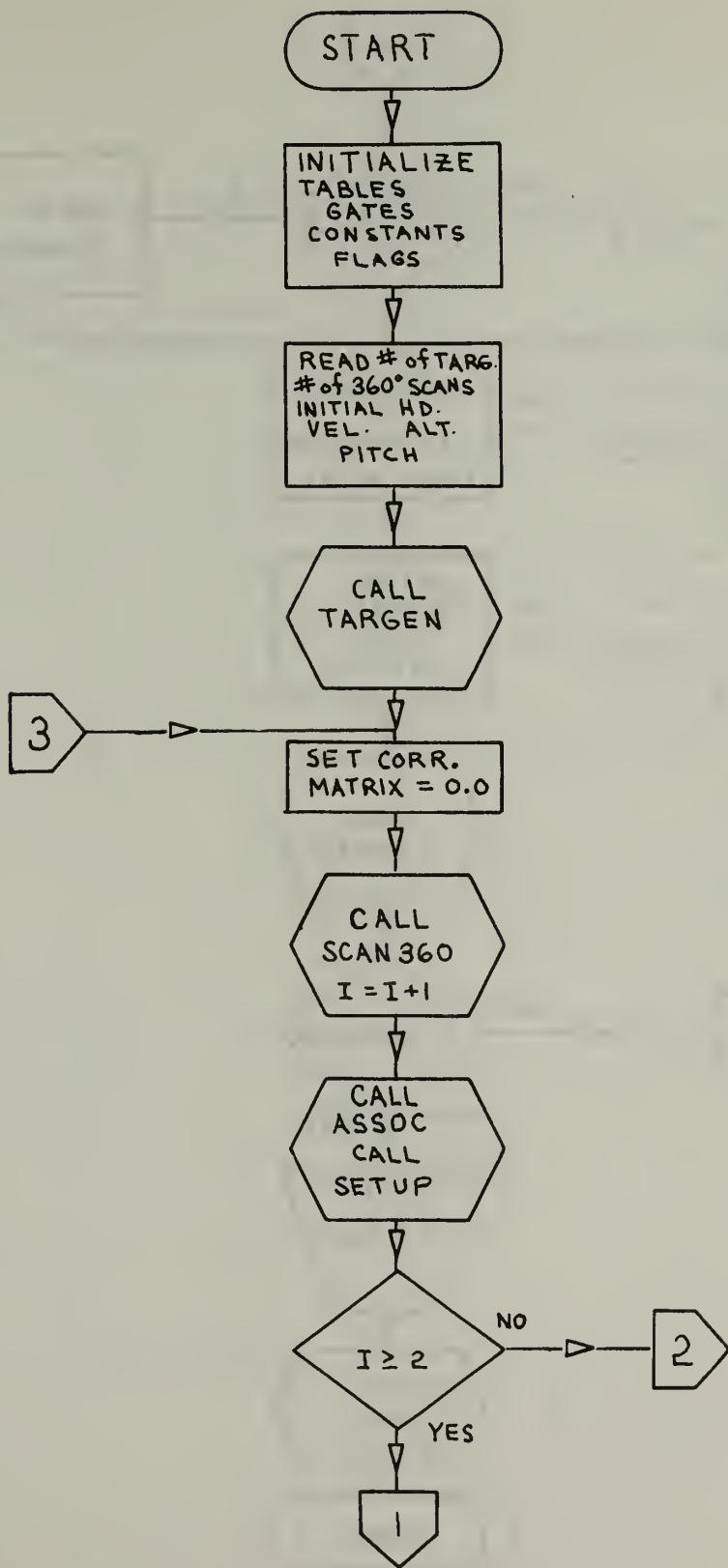


Decision.

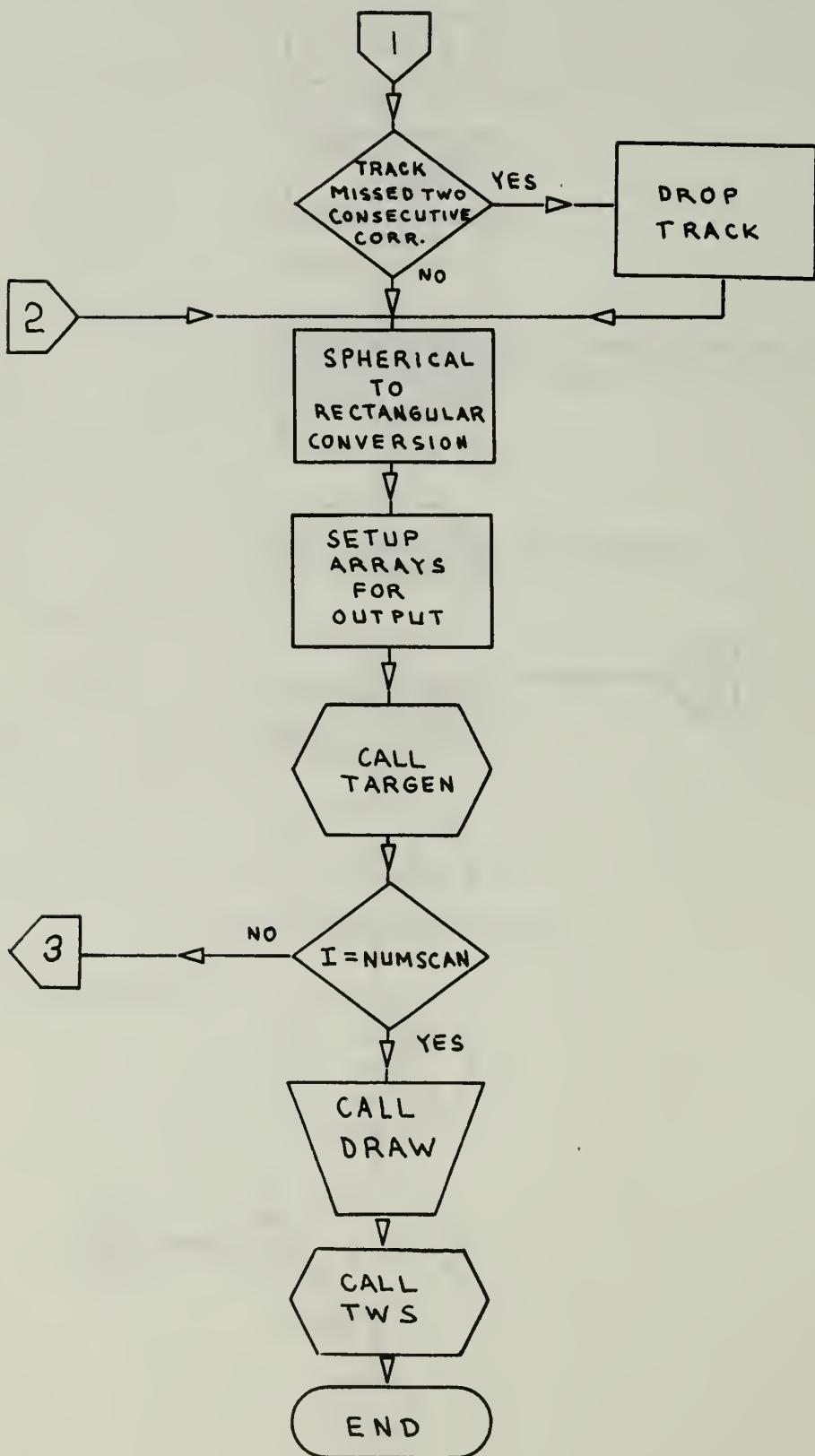


Processing, annotation.

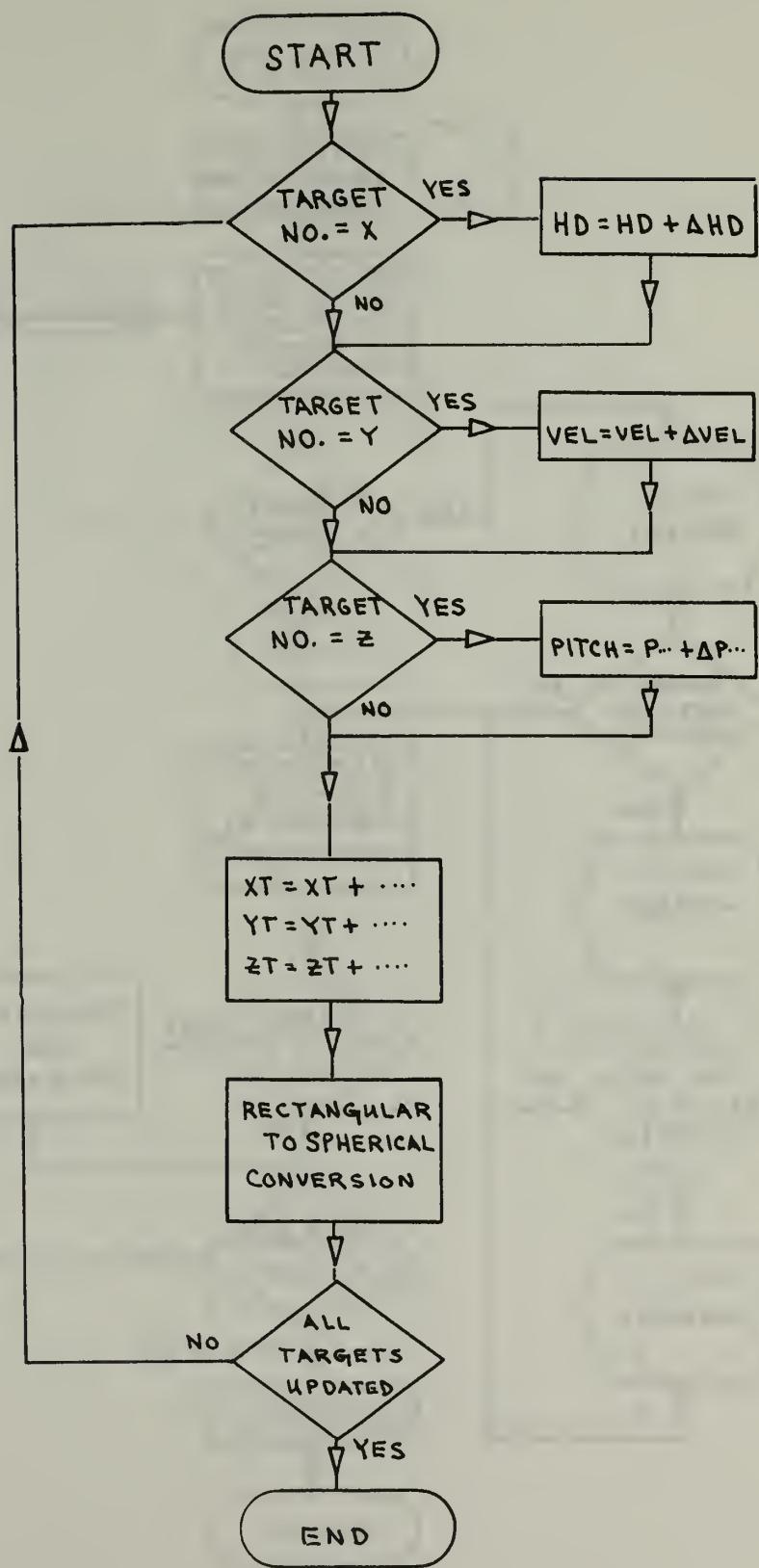
MAIN PROGRAM



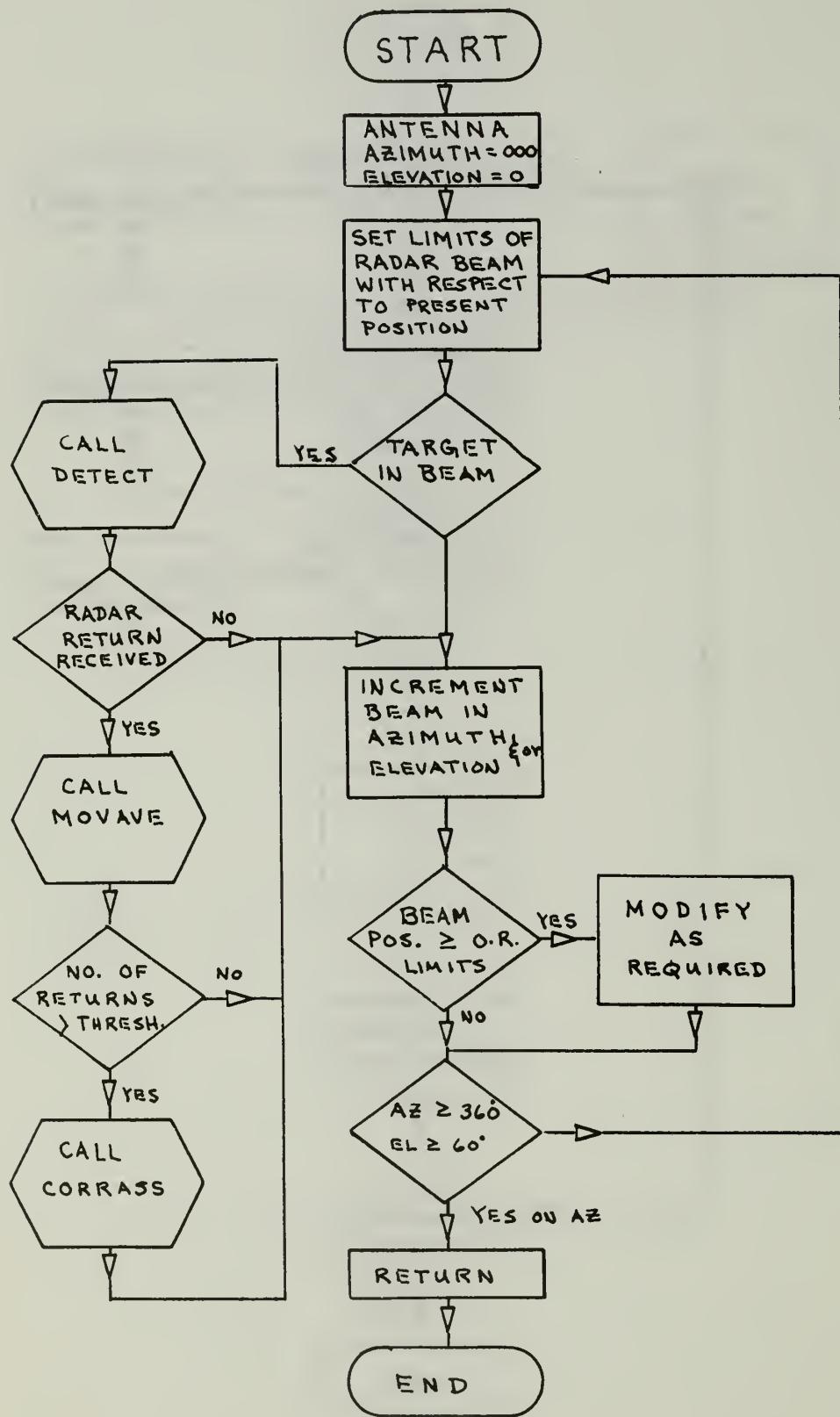
MAIN PROGRAM



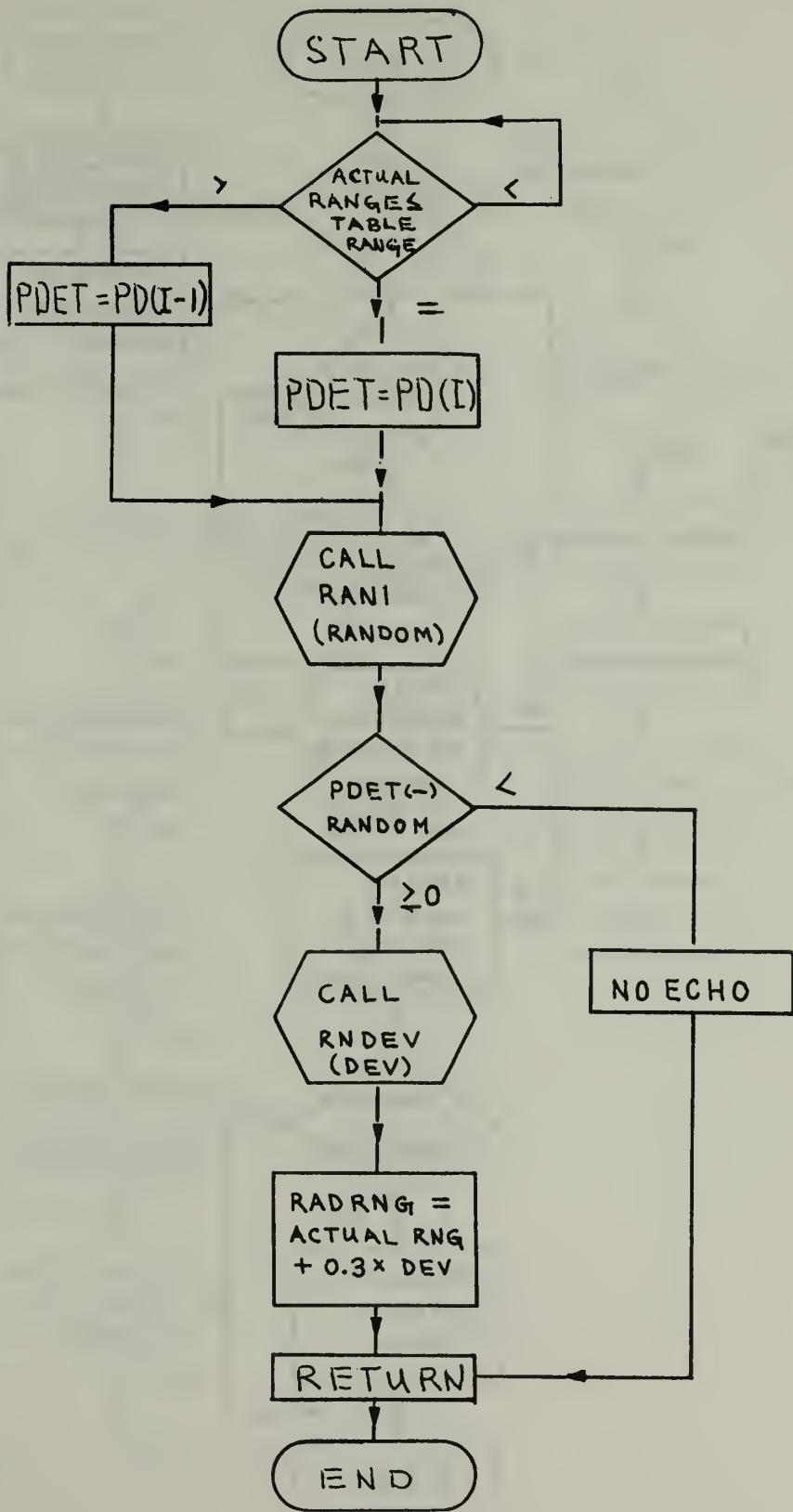
TARGEN



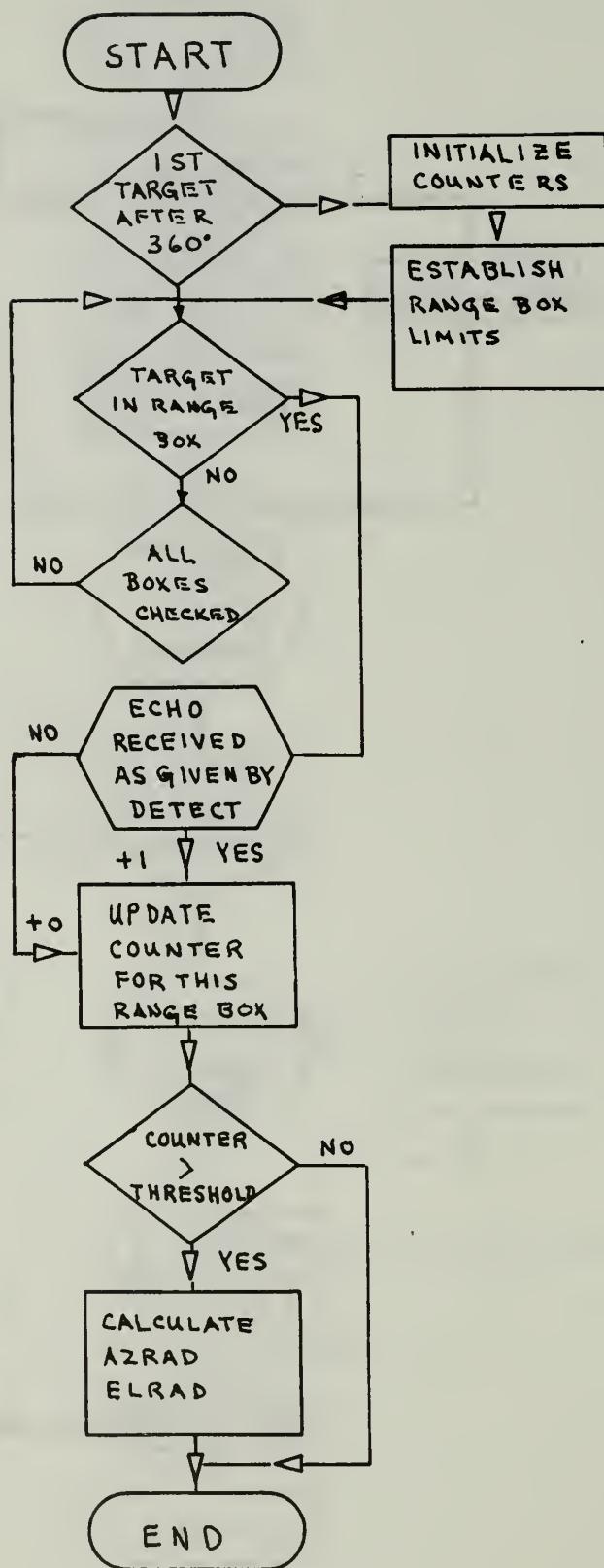
SCAN360



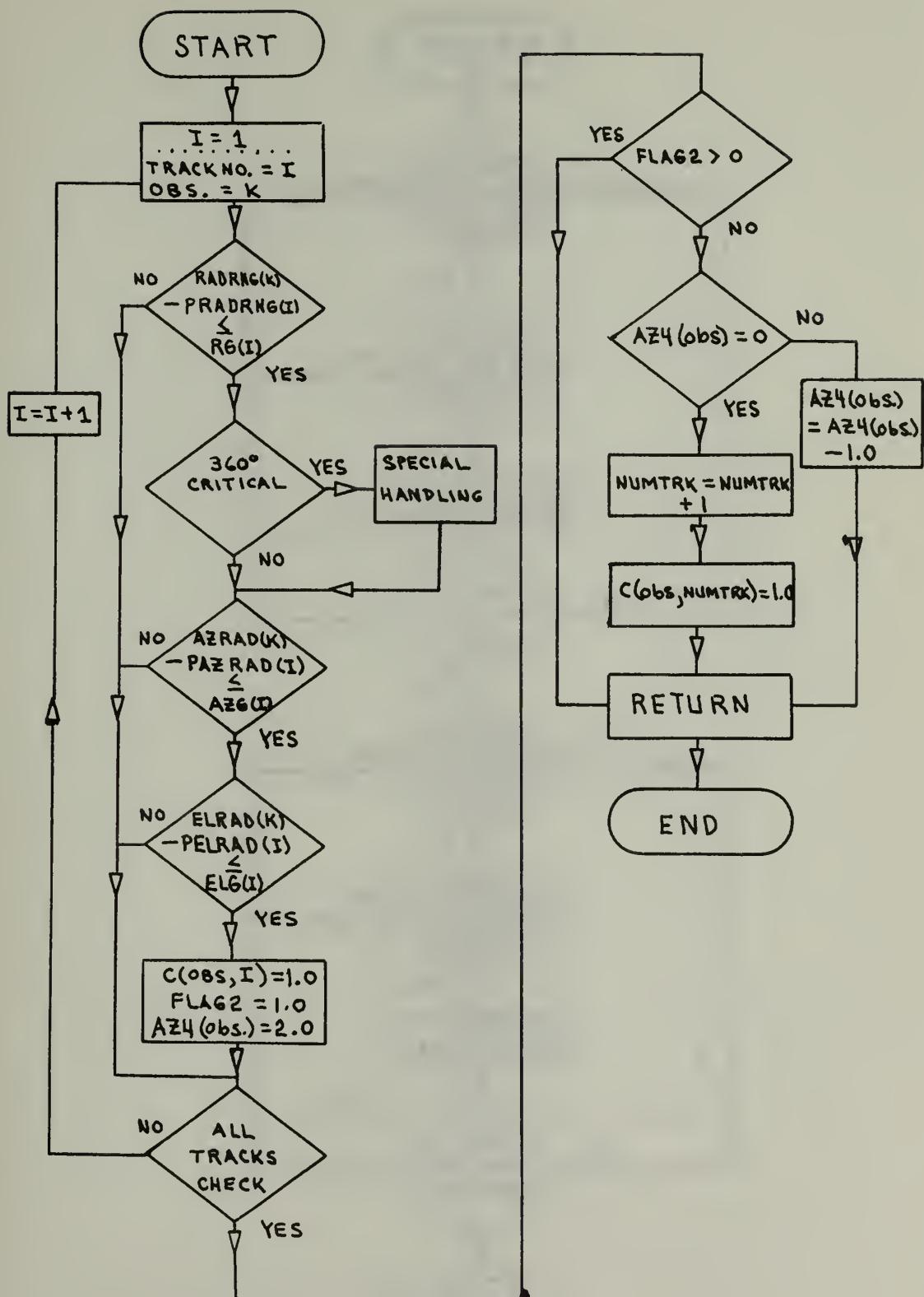
DETECT



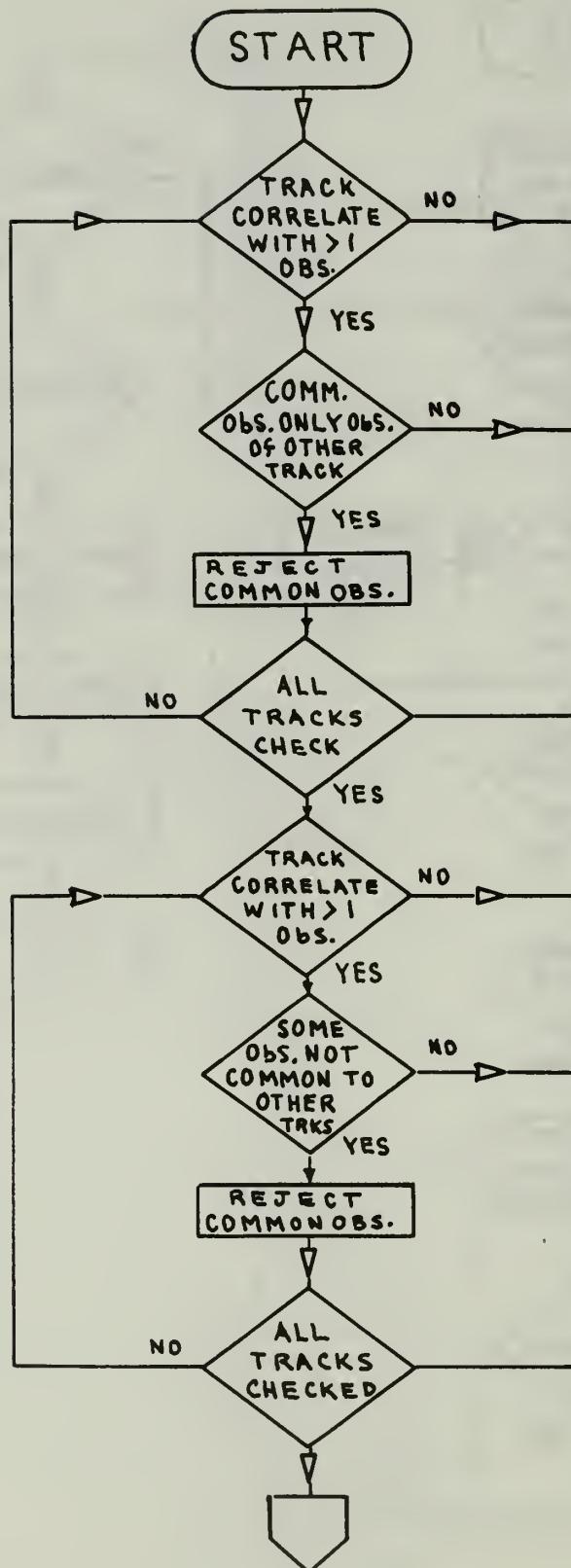
MOVAVE



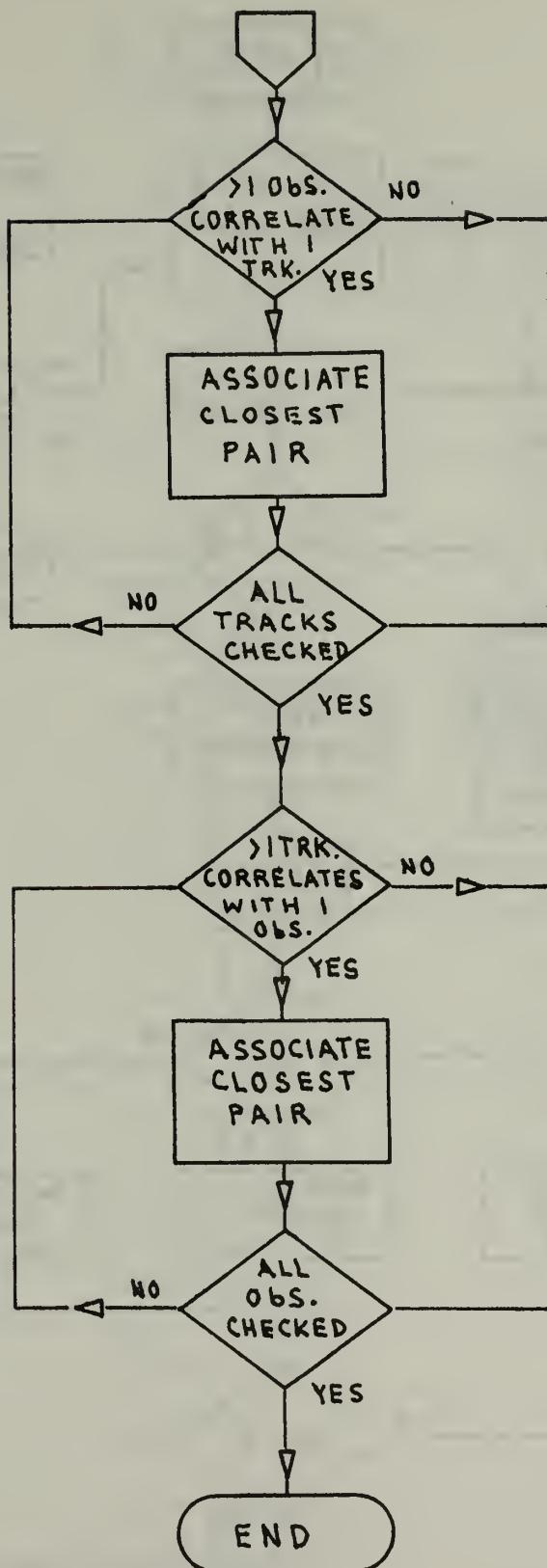
CORRASS



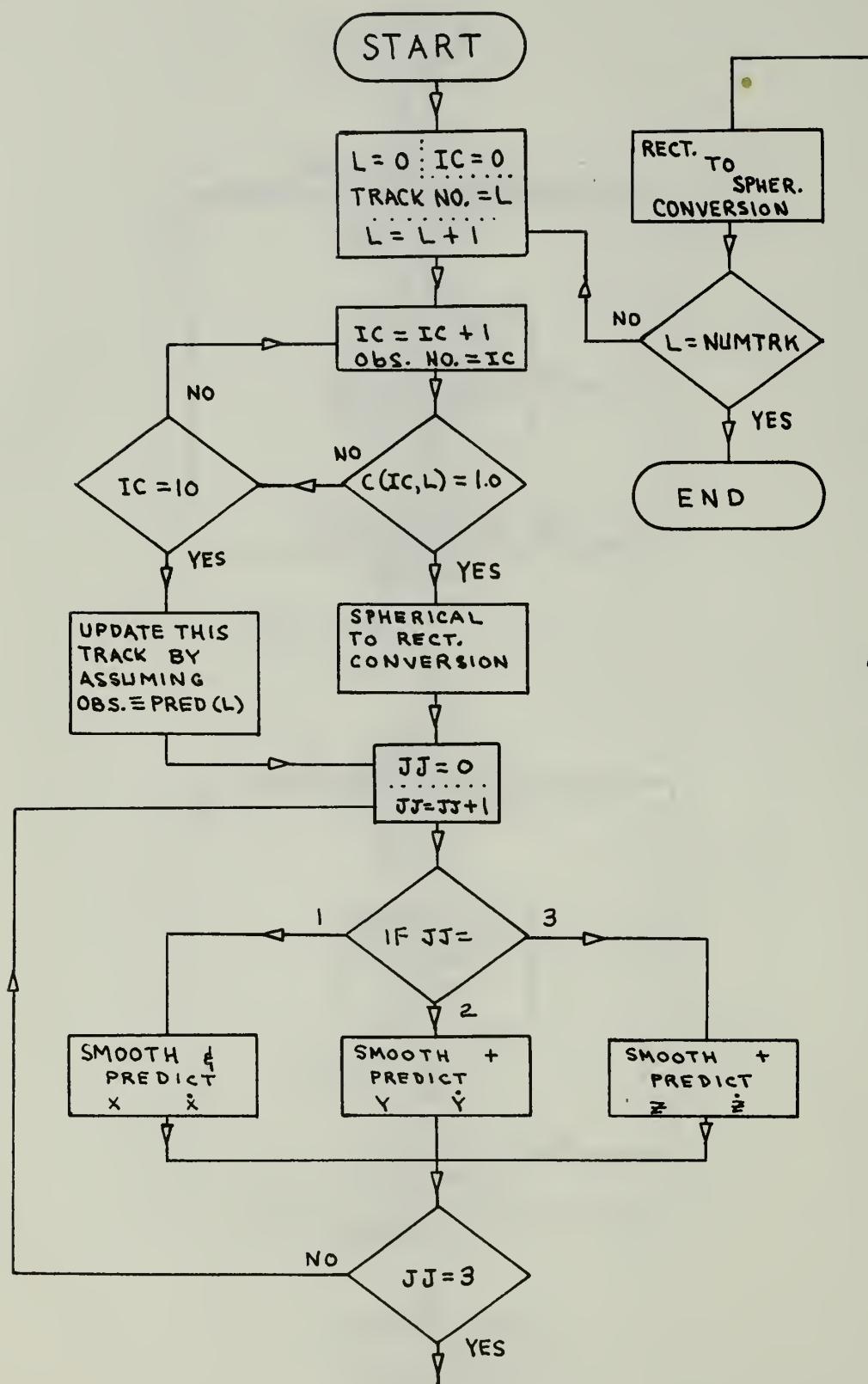
ASSOC



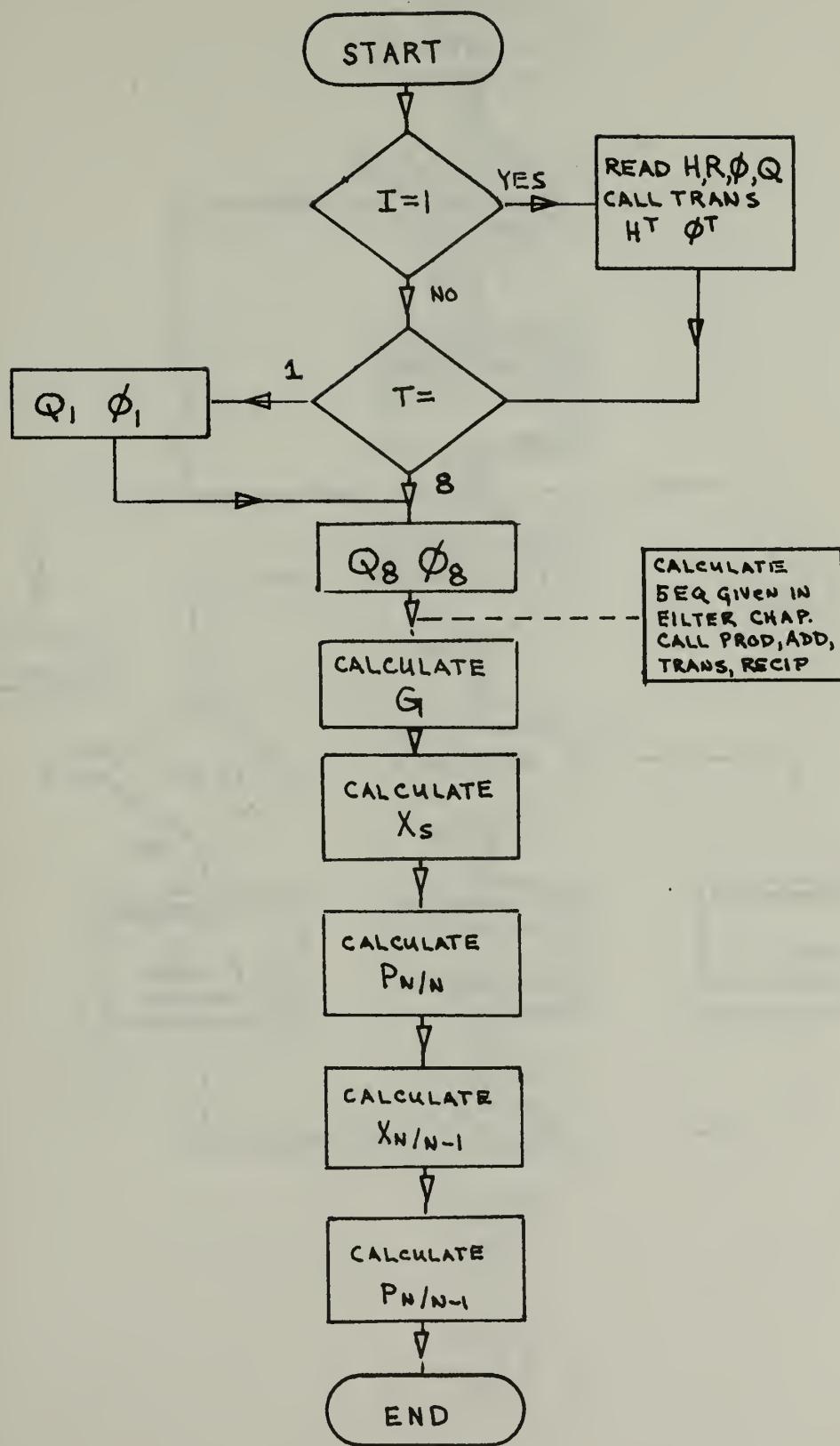
ASSOC (CONT)



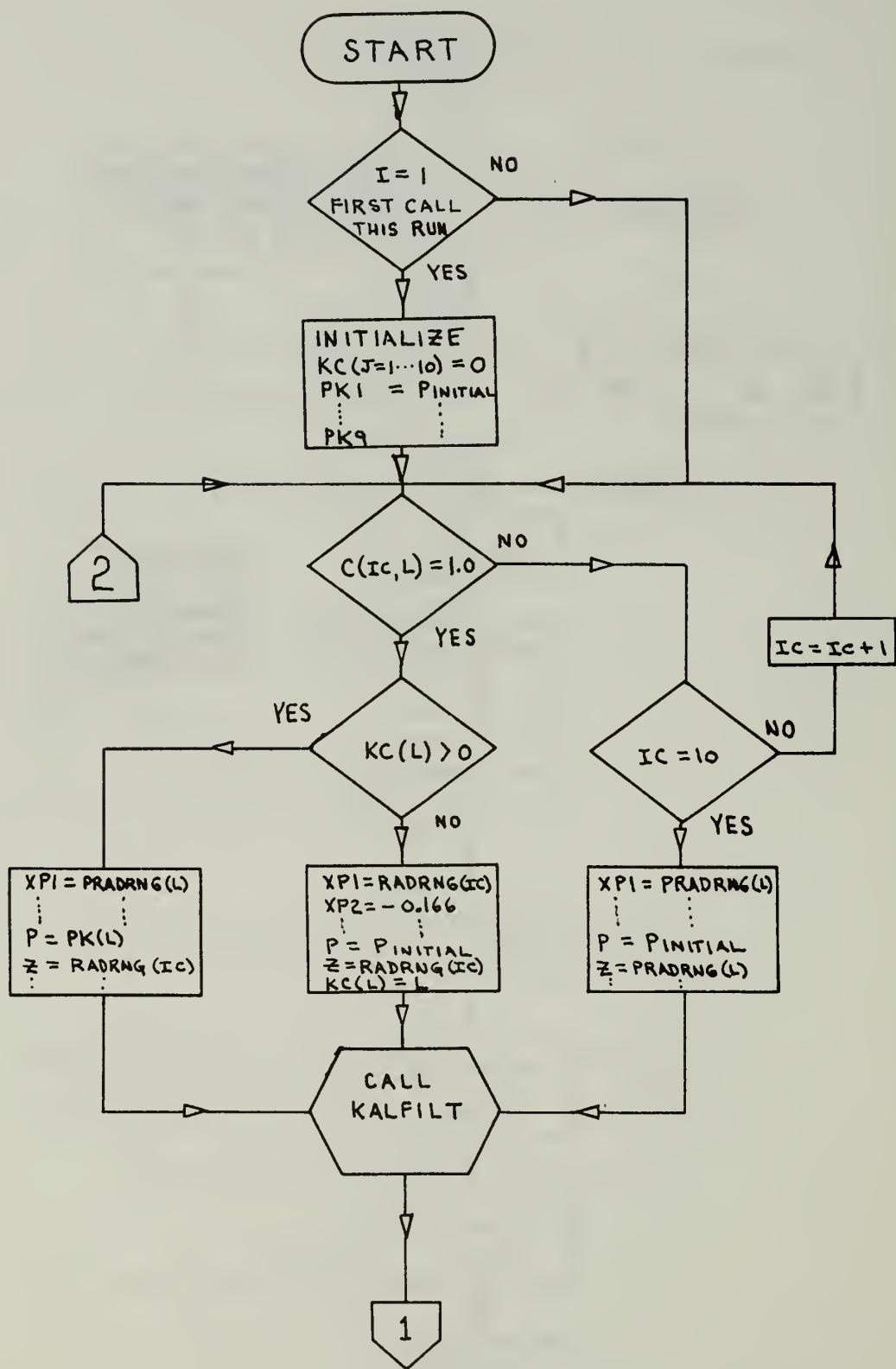
ABFILT



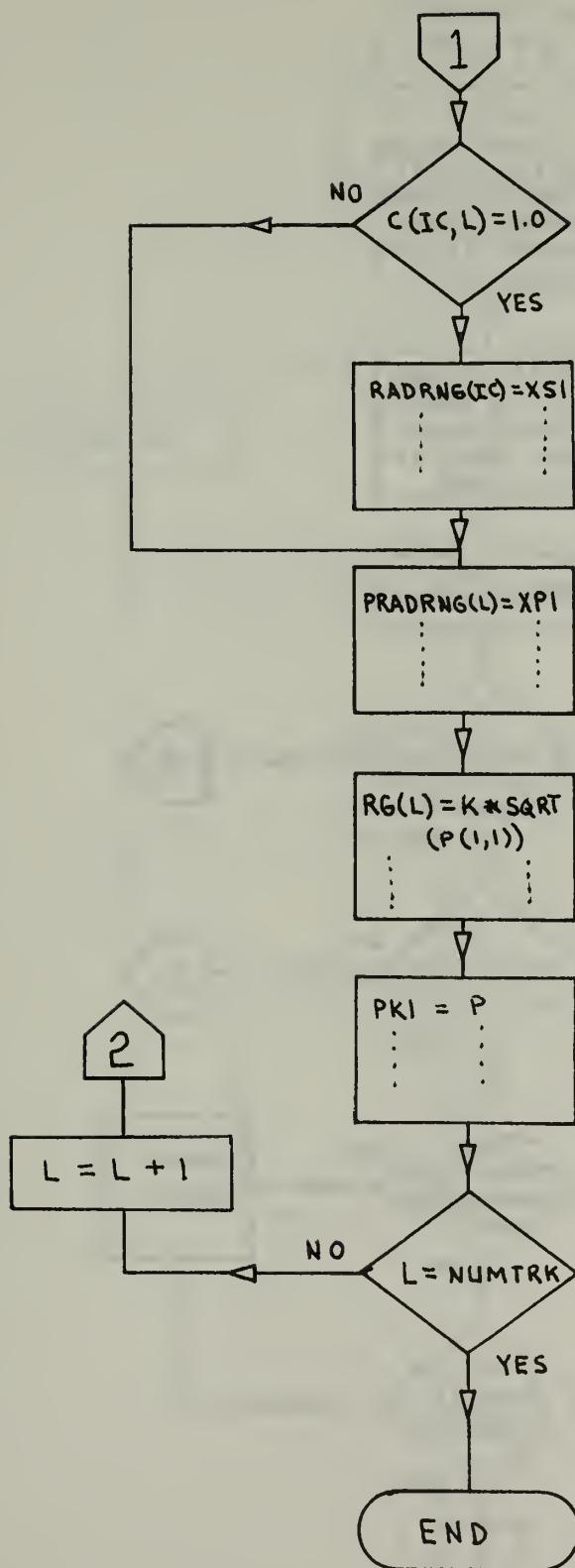
KALFILT



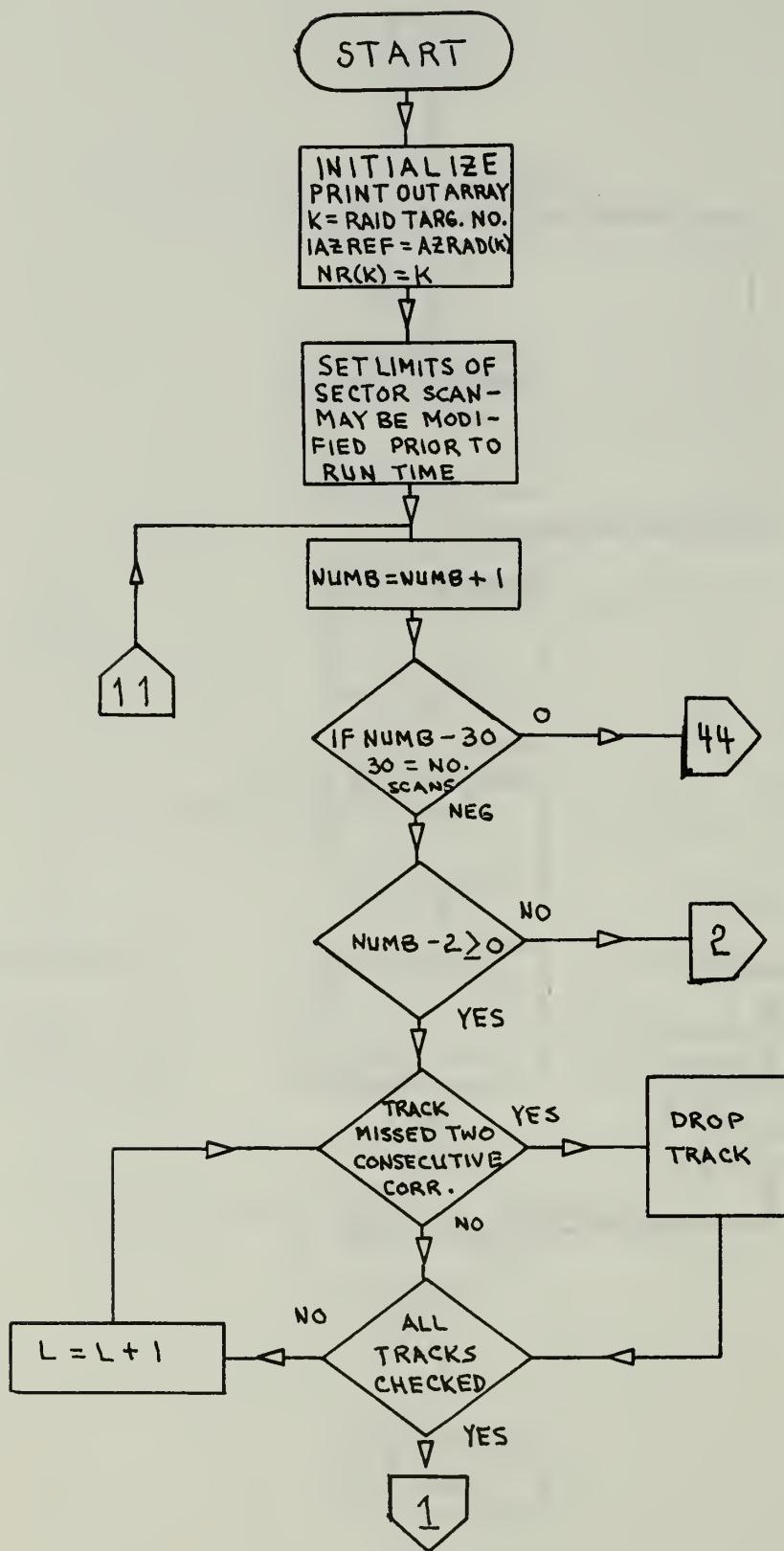
SETUP

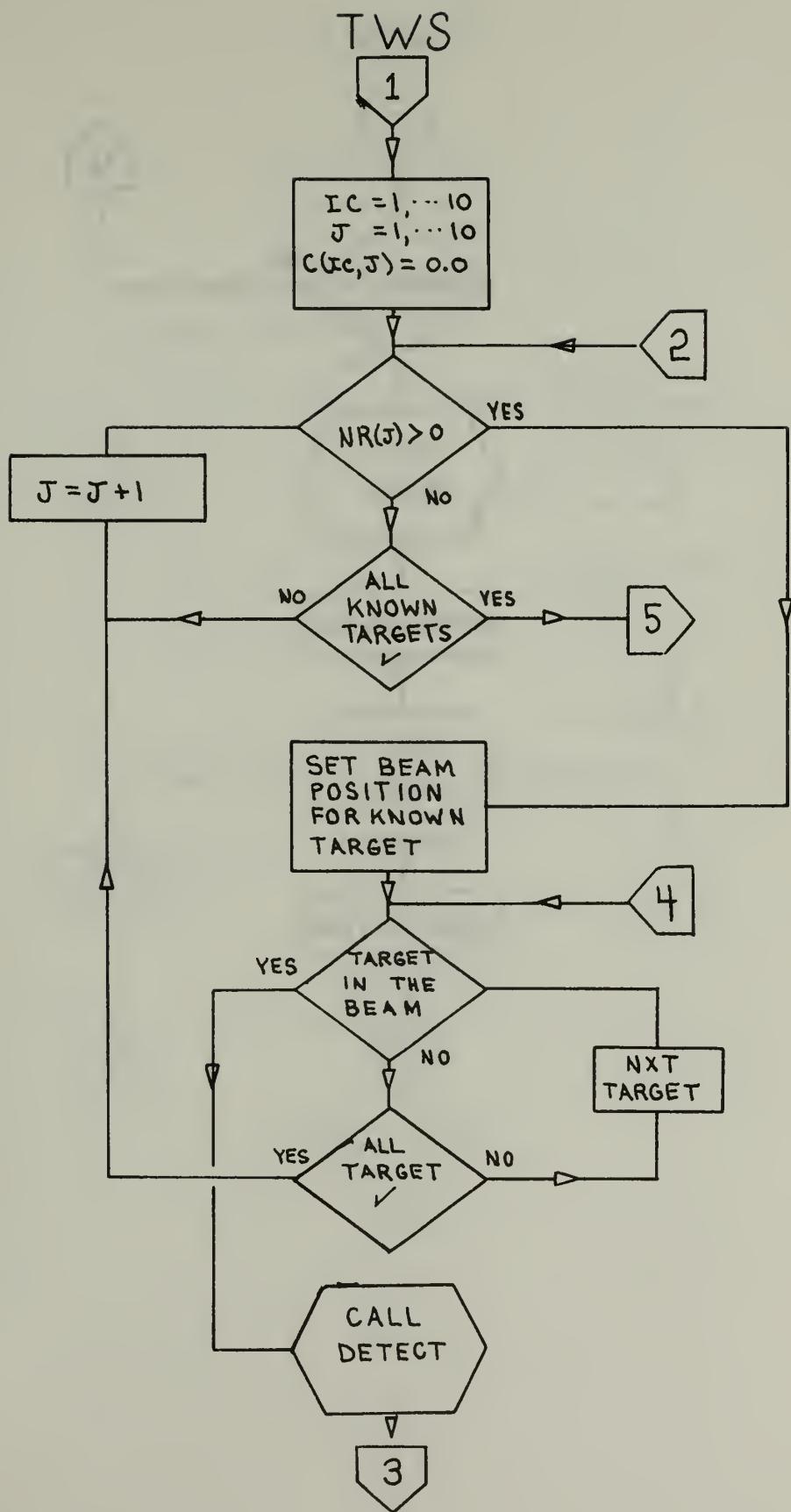


SETUP (CONT)

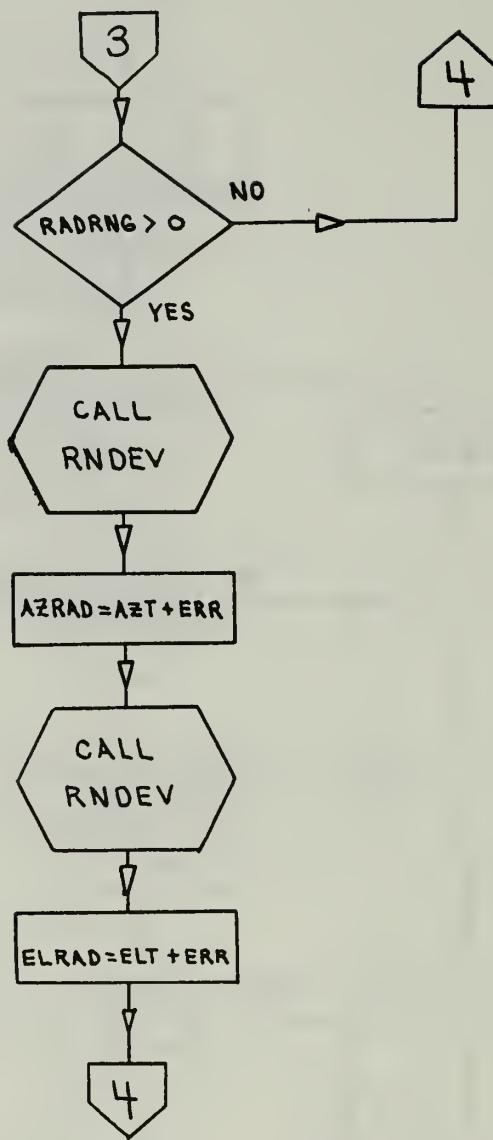


TWS

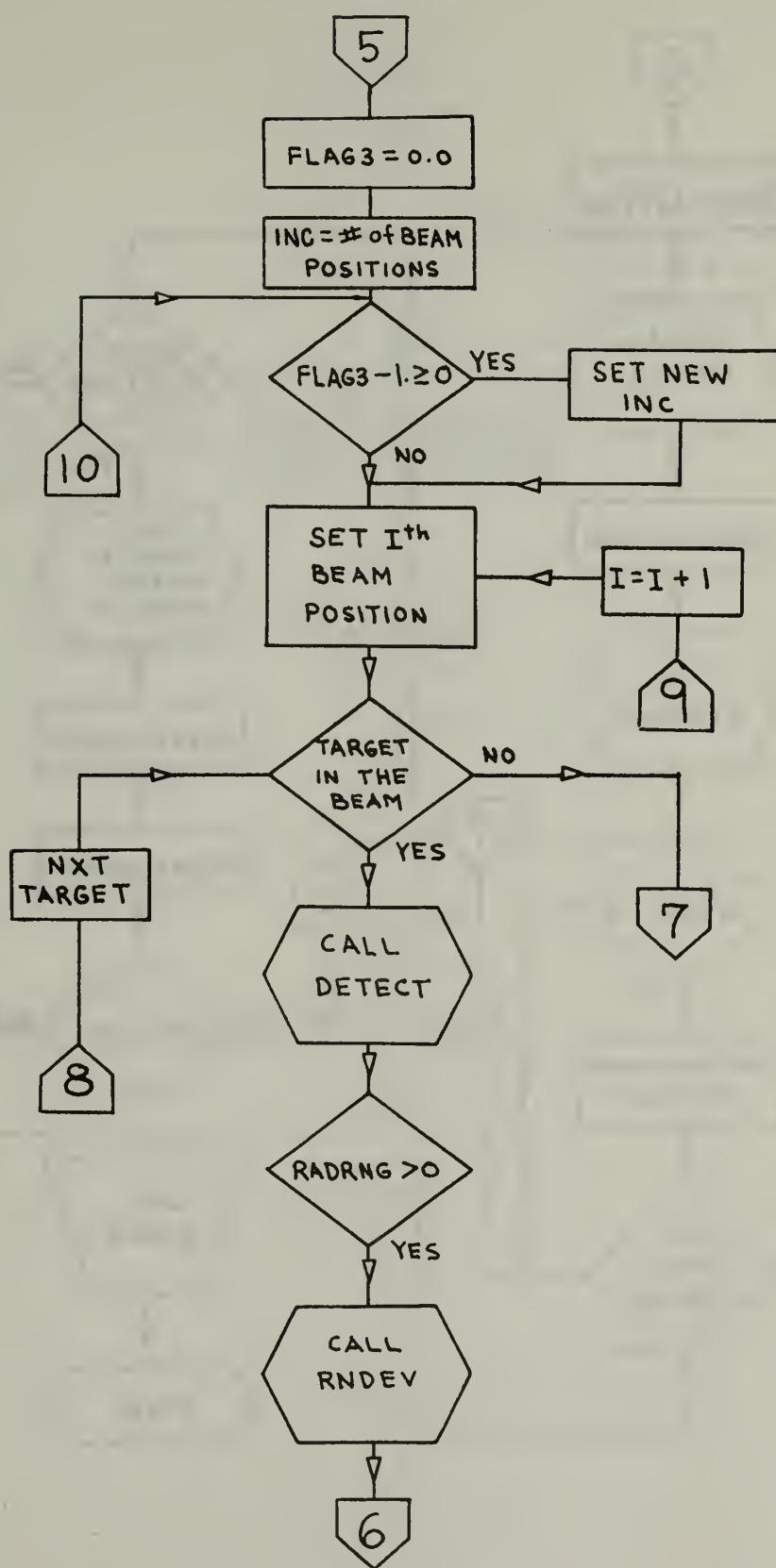




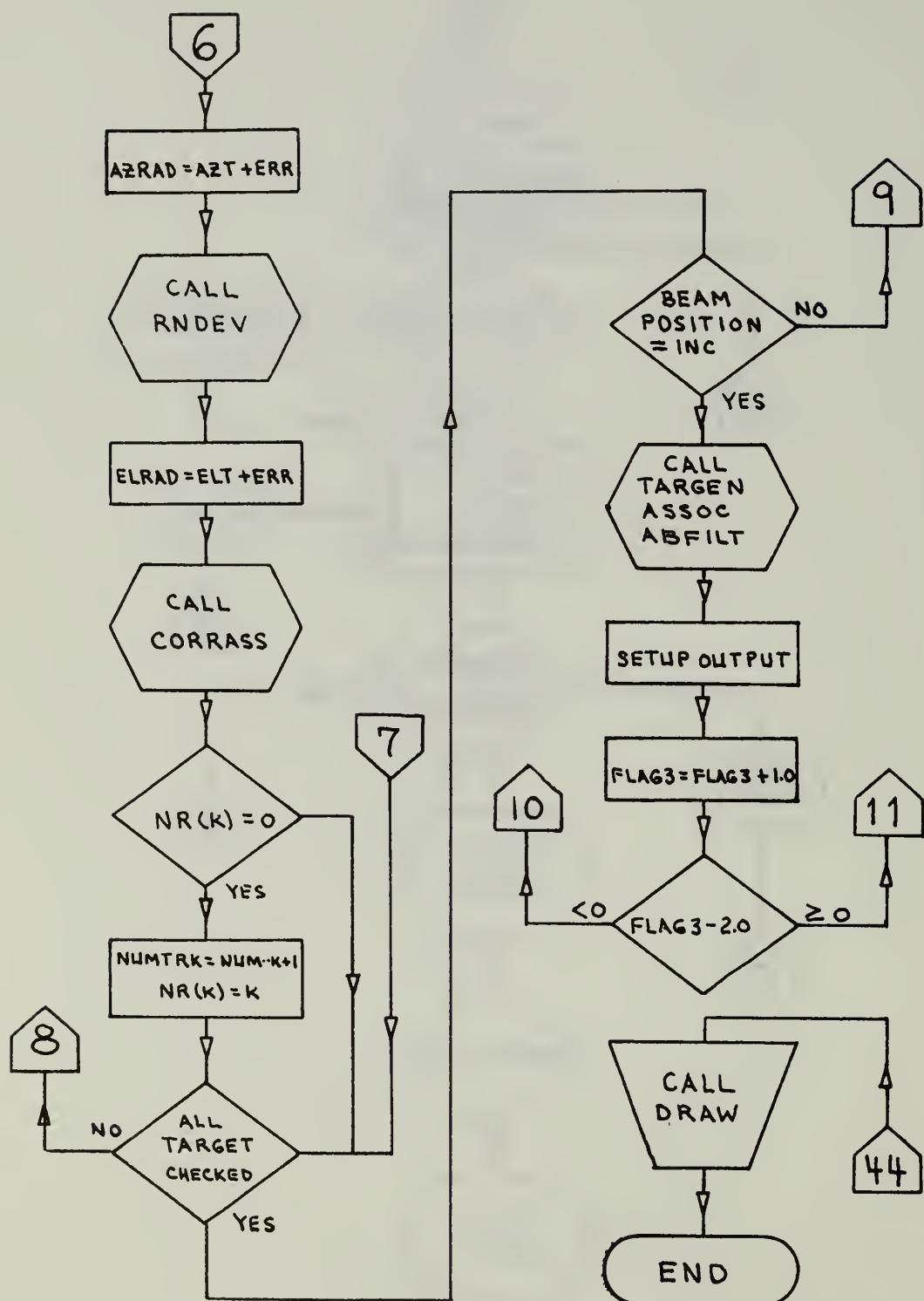
TWS



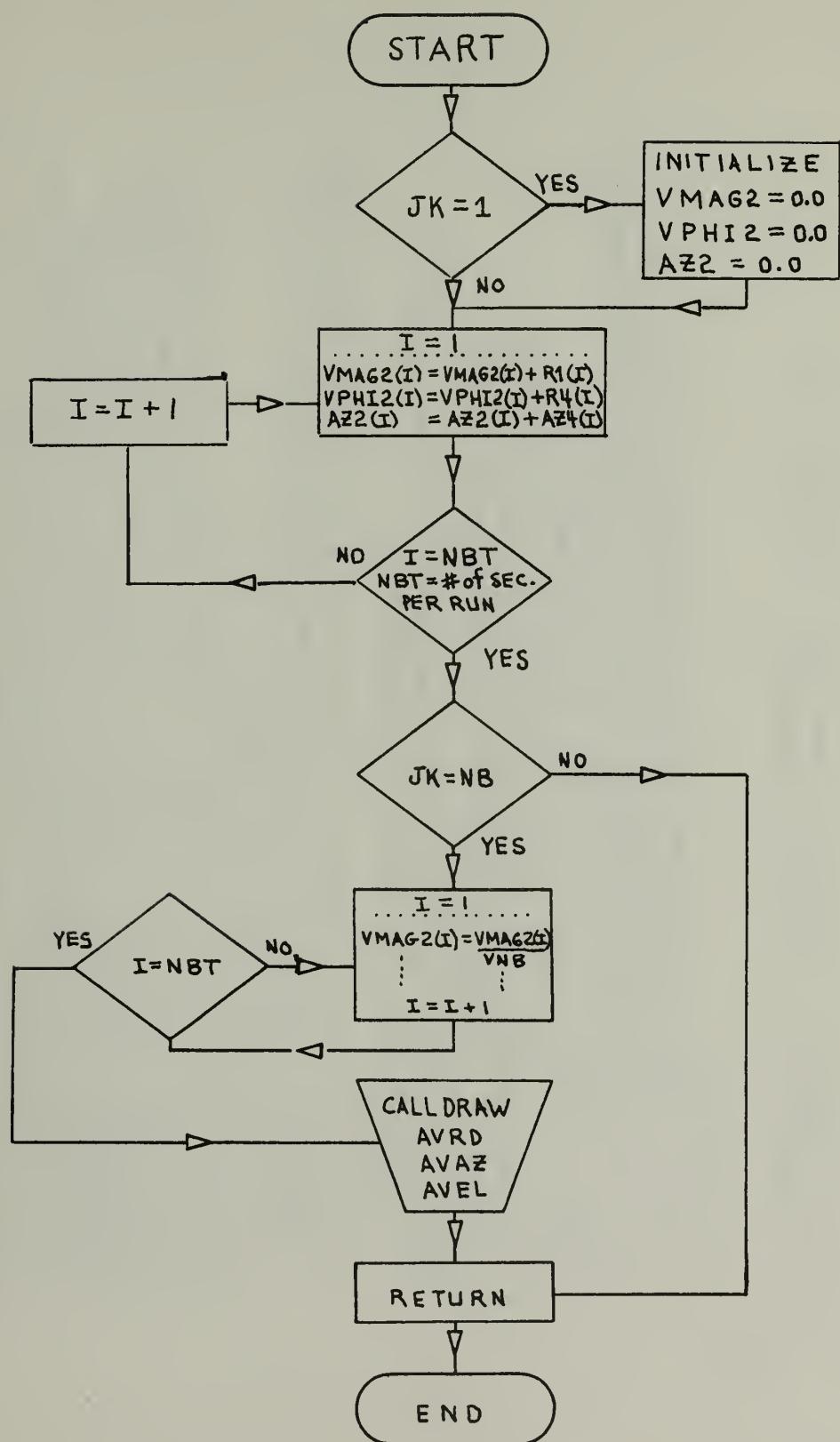
TWS



TWS



MONTE



APPENDIX IV
PROGRAM LISTING

PROGRAM KA21VGS

```

DIMENSION RANGE(10),TARGRNG( 10 ),PD(10),
1 LTITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10 ),YT( 10 ),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),YRAD(10),ZRAD(10),
3 ZELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRNG(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(200),YT1(200),
5 XRADI(200),YRADI(200),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
6 (10),PYRAD(10),PZRAD(10),DOTT(10),DCTZ(10),DOTY(10),NR(10),HDFIX(
7 10),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8 R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
9 ,XT2(30),YT2(30),XRAD2(30),YRAD2(3 )
DIMENSION HPHT(6,6),X(6,6),VN1(6,6),Z(6,6),PHT(6,6),
1 PHIT(6,6),P1(6,6),H(6,6),HT(6,6),PH1(6,6),R(6,6),Q(6,6),
2 XS(6,6),XP(6,6),XP1(6,6),T1(6,6),T2(6,6),Q1(6,6),Q8(6,6),PHI1(6,6),
3 ,PHI18(6,6)

DIMENSION RDOT(10),AZDOT(10),ELDOT(10),
1 KC(10),PK1(6,6),PK2(6,6),PK3(6,6),PK4(6,6),PK5(6,6),PK6(6,6),
2 PK7(6,6),PK8(6,6),PK9(6,6)
COMMON RANGE,TARGRNG,PDET,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TESTT,AZRAD1,
3 RADRNG1,ELRAD1,AZRAD ,C,S,DOT,T,PAZRAD,ELBEAMT
4 ,FLAG, NUMTRK, NUM1 ,NUM2,NUM3,AZBEAMT,RADDEG,ELRAD
5 ,XRAD,YRAD,ZRAD,CONT,PRADRNG,PELRAD,XT1,YT1,XRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DTY,NR,HDFIX,PITFIX,VELFIX
7 ,ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,AZ1,AZ2,AZ3,AZ4
8 ,XS,XP,P,KC

```

C
 C KALMAN FILTER IN POLAR COORDINATES
 C HEADING = DEGREES VELOCITY = N.M./HR. ALTITUDE = FEET

FORMAT STATEMENTS

```

100 FORMAT(4 F10.4)
101 FORMAT( /,3X,11H HEADING = ,F10.4, 7HDEGREES, 3X,10HVELOCITY = ,
1F10.4, 8HN.M./HR.3X,11HALITUDE = ,F10.4, 4HFEET,3X, 7HPITCH = ,
2F10.4, 7HDEGREES )
102 FORMAT( 2F10.4)
105 FORMAT(10X, F10.4 )
106 FORMAT(10F7.3 )
108 FORMAT( /, 9X,11HPROB OF DET, 9X,12HTARGET RANGE.,,10X,F10.4,
110X, F10.4 )
109 FORMAT( /,10X,F10.4,10X,F10.4)
110 FORMAT(10X,
110 FORMAT(10X,
120 FORMAT( F10.4)
122 FORMAT( /, 15,1X,6F10.4,15 )
123 FORMAT( /,4X,3H K ,3X, 8H RADRNG ,2X,6H AZRAD, 4X, 6H ELRAD, 4X,
15H XRAD , 5X, 5H YRAD , 5X, 5H ZRAD, 5X, 4H 1 )
161 FORMAT( /,10X,5HLL = ,12,5X,6HXT1 = ,F10.4,5X, 6HYT1 = , F10.4 )
162 FORMAT( /,10X, 5HLL = ,12,5X,8HXRAD1 = ,F10.4,5X, 8HYRAD1 = ,F10.4)
191 FORMAT( /,5X,5H K = ,12, 5X, 7H AZT = ,F10.4,5X,7H ELT = ,F10.4,
15H XT = ,F10.4, 5HYT = , F10.4 ,5X,10HTARGRG = ,F10.4)
192 FORMAT( /,5X,5H K = ,12, 5X, 7H AZT = ,F10.4,5X,7H ELT = ,F10.4,
15H XT = ,F10.4, 5HYT = , F10.4 ,5X,10HTARGRG = ,F10.4)

CONSTANTS AND FLAGS

RADDEG = 57.2957795131
NUNIF = 12207003125.
IR=71625307
READ 106,(PD(J),J=1,10)
READ 106,(RANGE(J),J=1,10)
PRINT 108,PD(1),RANGE(1)
DO 3 J=2,10
3 PRINT 109, PD(J),RANGE(J)
READ 110, NT
READ 110, NUMSCAN
LL=DO 4 K=1,NT
AZ4(K)=0.0

```

U U U U

```

RG(K) =2.5
AZG(K) = 2.5
ELG(K)=1.5
READ 100, HD(K), VEL(K), ALT(K), PITCH(K)
PRINT 101, HD(K), VEL(K), ALT(K), PITCH(K)
4 READ 102, XT(K), YT(K)
DO 2 J=1,12
2 ITITLE(J) = 8H
DO 53 J=1,10
INDEX(J)=0
53 INDEX(J)=0
T=8.0
NUM1=0
NUMTRK = 0
CORR=0.0
TESTT=1.0
PRINT 8
8 FORMAT(//,20X,37HCOVARIANCE OF THE ERROR OR P MATRIX )
CALL READD(6,6,P)
CALL TARGEN
DO 1 I=1,NUMSCAN
TEST = 0.0
JA=1
DO 55 IC=1,10
DO 55 J=1,10
55 C(IC,J)=0
CALL SCAN360
IF(I - 1) 61,61,60
60 NUM1 = 1
CALL ASSOC
61 CALL SETUP
IF(I-2)160,158,158
158 DO 150 J= 1,NUMTRK
TOTAL = 0.0
DO 151 IC = 1,10
151 TOTAL =TOTAL+ C(IC,J)

```

```

IF(TOTAL) 152,152,150
152 IF(INDEX(J)) 156,156,153
153 IF(I-(INDEX(J)+1)) 156,154,156
156 INDEX(J) =1
INDEX(J) = 1
GO TO 150
154 DO 155 JB=J,NUMTRK
BO 157 I=1,10
C((IC,JB))= C((IC,JB+1))
PRADRNG(JB)=PRADRNG(JB+1)
PAZRAD(JB)=PAZRAD(JB+1)
PELRAD(JB)=PELRAD(JB+1)
INDEX(JB)=INDEX(JB+1)
INDEX(JB)=INDE(JB+1)
INDE(JB)=PELRAD(JB+1)
157 PELRAD(JB)=PELRAD(JB+1)
155 CONTINUE
121 FORMAT(//,10X,11HDROP TRACK ,12)
PRINT 121,J
NUMTRK=NUMTRK -1
150 CONTINUE
160 DO 50 K= 1,NT
XRAD(K)=(RADRNG(K))*COSF( ELRAD(K )/57.29577) * *
1SINF(AZRAD(K )/ 57.29577)
YRAD(K)=(RADRNG(K))*COSF(ELRAD(K )/57.29577) *
1COSF(AZRAD(K )/ 57.29577)
LL= I
IF(K-1)200,200,201
200 XT1(LL)= XT(K)
YT1(LL)= YT(K)
IF(I-1) 57,57,58
57 XRAD1(LL)=XRAD(K)
YRAD1(LL)=YRAD(K)
GO TO 56
58 CC= •0
DO 51 J=1,10
51 CC=CC+C(K,J)

```

```

1 IF(CC) 54,54,52
52 XRADI(LL)=XRAD(K)
YRADI(LL)=YRAD(K)
GO TO 56
54 XRADI(LL)=0.0
YRADI(LL)=0.0
GO TO 56
201 XT2(LL)=XT(K)
YT2(LL)=YT(K)
IF(I-1) 557,557,558
557 XRAD2(LL)=XRAD(K)
YRAD2(LL)=YRAD(K)
GO TO 56
558 CC= 0
BO 551 J=1,10
551 CC=CC+C(K,J)
IF(CC) 554,554,552
552 XRAD2(LL)=XRAD(K)
YRAD2(LL)=YRAD(K)
GO TO 56
554 XRAD2(LL)= 0.0
YRAD2(LL)= 0.0
56 PRINT 123
PRINT122, K,RADRN(K),AZRAD(K), ELRAD(K),XRAD(K), YRAD(K)
1), ZRAD(K), 1
PRINT 161,LL,XT1(LL),YT1(LL)
PRINT 162,LL, XRADI(LL),YRADI(LL)
50 CONTINUE
CALL TARGEN
1 CONTINUE
ITITLE(1)= 8HFULL SCA
ITITLE(2)=8HN MANUEV
ITITLE(3)=8HING TARG
ITITLE(4)=8H3 DEG/SE
ITITLE(5)=8HC
ITITLE(6)=8H

```

```

ITITLE( 7)=8H  DELANEY
ITITLE( 8)=8H  W.F.
ITITLE( 9)=8HKALMAN
ITITLE(10)=8HJOB0194
ITITLE(11)=8H
ITITLE(12)=8H
LABEL=4H
CALL DRAW(LL,XT1,YT1,1,0,LABEL,ITITLE, 3.,0,0,2,2,5, 8,0, LAST)
CALL DRAW(LL,XT2,YT2,2,0,LABEL,ITITLE, 3.,0,0,2,2,5, 8,0, LAST)
CALL DRAW(LL,XRAD2,YRAD2,2,5,LABEL,ITITLE, 3.,3.,0,3,2,2,6,10,0,
1LAST)
CALL DRAW(LL,XRAD1,YRAD1,3,1,LABEL,ITITLE, 3.,3.,0,3,2,2,6,10,0,
1LAST)
CALL TWS
END

SUBROUTINE SCAN360
DIMENSION  RANGE(10)•TARGRNG( 10 ),PD(10),
ITITLE(12)•HD(10)•VEL(10)•ALT(10)•PITCH(10)•XT( 10 )•YT( 10 ),
22T(10)•AZT(10)•C(10,10)•S(10)•ELT(10)•XRAD(10)•YRAD(10)•ZRAD(10),
3ELRAD(10)•AZRAD(10)•BOXMIN(10)•BOXMAX(10)•ELTEST(10)•IAZ(10),
4PRADRNG(10)•PELRAD(10)•PAZRAD(10)•RADRNG(10)•XT1(200)•YT1(200),
5XRAD1(200)•YRAD1(200)•INDEX(10)•INDE(10)•CONT(10)•IAT(10)•PXRAD
6(10)•PYRAD(10)•PZRAD(10)•DOTX(10)•DOTZ(10)•DOTY(10)•NR(10)•HDFIX(
710)•PITFIX(10)•VELFIX(10)•RG(10)•AZG(10)•ELG(10)•R1(20)•R2(20),
8R3(20)•R4(20)•AZ1(20)•AZ2(20)•AZ3(20)•AZ4(20)
DIMENSIONXS(6,6)•XP(6,6)•P(6,6)•KC(10)
COMMON  RANGE•TARGRNG•PDET•PD•IR•RANDOM•NUNIF•DEV,
1 RANERR, RDRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT , ELT , IAZ , BOXMAX,DOXMIN,ELTEST,TESTT,AZRAD1,
3RADRNG1,ELRAD1, AZRAD , C,S,DOTT,ZT,PAZRAD,ELBEAMT
4•FLAG, NUMTRK, NUM1 , NUM2,NUM3,AZBEAMT,RADDEG,ELRAD
5• XRAD, YRAD,ZRAD,CONT,PRADRNG,PELRAD,XT1,YT1•XRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DOTY,NR,HDFIX,PITFIX,VELFIX
7 •ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4
8•XS,XP,P,KC
GO= .0

```

```

SUM=0.00
8  II=1
    ELBEAMB=0.0
    ELBEAMT=5.454545
11  AZBEAML = SUM
    AZBEAMT = SUM -1.0
    NA = 0
    IF(AZBEAMT )60,61,61
60  AZBEAMT =360.0+AZBEAMT
    NA = 1
61  DO 63 K=1,NT
    IF(NA) 12,13,12
12  IF(AZT(K) -1.0) 21,20,20
20  IF(AZT(K) - AZBEAMT)63,63,5
21  IF(AZT(K) - AZBEAML)5,63,63
13  IF(AZT(K) -AZBEAML)4,63,63
4   IF(AZBEAMT - AZT(K))5,63,63
5   IF(ELT(K) -ELBEAMB)63,6,6
6   IF(ELBEAMT - ELT(K))63,7,7
7   NUM3 = K
    CALL DETECT
    CALL MOVAVE
    IF(FLAG) 63,63,52
52  NUM3 = K
    CALL CORRASS
63  CONTINUE
    II=II+1
    SUM = SUM + .02
    ELBEAMB = ELBEAMT
    ELBEAMT=ELBEAMT+5.454545
    IF(GO -1.0) 15,16,16
15  IF( 90.0-SUM) 14,14,10
14  SUM=270.0
    GO=1.0
    GO TO 10
16  IF( 360.0- SUM) 9,9,10

```

```

10 IF(II-11) 11,8,8
9 GO =0.0
RETURN
END
SUBROUTINE DETECT
DIMENSION  RANGE(10),TARGRNG( 10  ),PD(10),
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10  ),YT( 10  ),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),ZRAD(10),
3 ELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRNG(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(200),YT1(200),
5 XRADI(200),YRAD1(200),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
6(10),PYRAD(10),PZRAD(10),DOTX(10),DOTZ(10),DOTY(10),NR(10),HDFIX(
710),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
DIMENSIONX(6,6),XP(6,6),P(6,6),KC(10)
COMMON  RANGE,TARGRNG,PDET,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT   , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TESTT,AZRAD1,
3 RADRNG1,ELRAD1, AZRAD, C,S,DOTT,ZT,PAZRAD,ELBEAMT
4,FLAG, NUMTRK, NUM1,NUM2,NUM3,AZBEAMT,RADDEG,ELRAD
5 ,XRAD,YRAD,ZRAD,CONT,PRADRNG,PELRAD,XT1,XT2,YRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DOTY,NR,HDFIX,PITFIX
7 ,ALPHA,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4
8 ,XS,XP,P,KC
K=NUM3
IF(RANGE(1)- TARGRNG(K ) ) 20,20,21
20 PDET= PD(1)
GO TO 5
21 DO 1 J=1,10
  IF(RANGE(J) -TARGRNG(K ) )2,3,1
  3 PDET=PD(J)
  GO TO 5
  2 PDET=PD(J-1)
  GO TO 5
  1 CONTINUE
  PDET=PD(10)

```

```

5 CALL RANI(IIR, RANDOM)
IF(PDET - RANDOM) 7,6,6
6 CALL RNDEVI(NUNIF, DEV)
RANERR = (DEV*0.3)
RADRNG(K) = TARGRNG(K) + RANERR
PRINT 412, K, RADRNG(K)
412 FORMAT(/,10X,4HK = ,12, 9HRADRNG = , F10.4,
1 20HA GOOD RADAR TARGET )
GO TO 8
7 RADRNG(K) =0.0
411 FORMAT(/,10X,13H TARGET NO. = , 12, 7H A MISS )
PRINT 411,K
8 RETURN
END

SUBROUTINE MOVAVE
DIMENSION          RANGE(10), TARGRNG( 10 ), PD(10),
1 ITITLE(12), HD(10), VEL(10), ALT(10), PITCH(10), XT( 10 ), YT( 10 ),
2 ZT(10), AZT(10), C(10,10), S(10), ELT(10), XRAD(10), YRAD(10), ZRAD(10),
3 ZELRAD(10), AZRAD(10), BOXMIN(10), BOXMAX(10), ELTEST(10), IAZ(10),
4 PRADRNG(10), PELRAD(10), PAZRAD(10), RADRNG(10), XT1(200), YT1(200),
5 XRAD1(200), YRAD1(200), INDEX(10), INDE(10), CONT(10), IAT(10), PXRAD
6(10), PYRAD(10), PZRAD(10), DOTX(10), DOTY(10), NR(10), HDFIX(710),
710, PITFIX(10), VELFIX(10), RG(10), AZG(10), ELG(10), R1(20), R2(20),
8R3(20), R4(20), AZ1(20), AZ2(20), AZ3(20), AZ4(20)
DIMENSION XS(6,6), XP(6,6), P(6,6), KC(10)
COMMON          RANGE, TARGRNG, PDET, PD, IR, RANDOM, NUNIF, DEV,
1 RANERR, RADRNG, HD, VEL, ALT, PITCH, XT, YT, NT, JA, T, I, TEST,
2 AZT, ELT, IAZ, BOXMAX, BOXMIN, ELTEST, TESTT, AZRAD1,
3 RADRNG1, ELRAD1, AZRAD, C, S, DOTT, ZT, PAZRAD, ELBEAMT
4, FLAG, NUMTRK, NUM1, NUM2, NUM3, AZBEAMT, RADDEG, ELRAD
5, XRAD, YRAD, ZRAD, CONT, PRADRNG, PELRAD, XT1, XRAD1, YRAD1, INDEX, IN
6 DE, IAT, PXRAD, PYRAD, PZRAD, DOTX, DOTZ, DOTY, NR, HDFIX, PITFIX, VELFIX
7, ALPHA, BETA, RG, AZG, ELG, R1, R2, R3, R4, AZ1, AZ2, AZ3, AZ4
8, XS, XP, P, KC
K=NUM3
FLAG = 0.0

```

```

IF (RADRNG(K) ) 7•1•7
1 RADRNG1=TARGRNG(K )
GO TO 2
7 RADRNG1 =' RADRNG(K )
2 IF (TEST) 3,4,3
4 IA=
DO 17 K=1,5
1AZ(K ) =0
17 IAT(K ) = 0
NO=
K = NUM3
TEST=1•0
10 NO = NO + 1
IA=IA+1
BOXMIN (IA)=RADRNG1-1•5
BOXMAX (IA) = RADRNG1 + 1•5
ELTEST (IA) = ELBEAMT
3 DO 5 J=1,NO
  IF ( ELTEST (J)-ELBEAMT ) 5,6,5
  6 IF ( BOXMAX(J) - RADRNG1 ) 5,8,8
  8 IF (RADRNG1 - BOXMIN(J) ) 5,9,9
5 CONTINUE
GO TO 10
9 IF (RADRNG(K ) )11,12,11
12 IAZ(K )=IAZ(K )+0
  IAT(K )=IAT(K )+1
GO TO 22
11 IAZ(K ) = IAZ(K ) +1
  IAT(K )=IAT(K )+1
  IF (IAZ(K )-1) 22,21,22
21 SONT(K ) =AZBEAMT+0•99
22 IF (IAT(K )-5) 14,16,14
  16 IF (IAZ(K )- 3) 20,15,15
  20 RADRNG(K ) = 0•0
GO TO 14
15 N = K

```

```

FLAG = 1.0
NUM3 = K
AZRAD(N) = CONT(K)-0.49
CALL RNDEV(NUNIF,DEV)
ELERR= DEV*0.5
ELRAD(N) = ELT(K) + ELERR
TESTT=TESTT +1.0
PRINT 18,AZRAD(N), ELRAD(N) , ELERR
18 FORMAT( /,10X,13HGO TO CORRASS ,8HAZRAD = ,F10.4, 10X, 8HELRAD = ,
1 F1 •4,10X,8HELERR = , F10.4)
14 RETURN
END
SUBROUTINE CORRASS
DIMENSION RANGE(10),TARGRNG( 10 ),PD(10),
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10 ),YT( 10 ),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),YRAD(10),ZRAD(10),
3 ELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRN(10),PELRAD(10),PAZRAD(10),RADRN(10),XT1(200),YT1(200),
5 XRAD1(200),YRAD1(200),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
6(10),PYRAD(10),PZRAD(10),DUTX(10),DOTZ(10),DOTY(10),NR(10),HDFIX(
710),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
DIMENSIONXS(6,6),XP(6,6),P(6,6),KC(10)
COMMON RANGE,TARGRNG,PDET,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRN,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TESTT,AZRAD1,
3 RADRN1,ELRAD1, AZRAD ,C,S,DOTT,ZT,PAZRAD,ELBEAMT
4,FLAG, NUMTRK, NUM1 ,NUM2,NUM3,AZBEAMT,RADDEG,ELRAD
5 ,XRAD •YRAD•ZRAD,CONT,PRADRN,PELRAD,XT1,XRAD1,YRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTZ,DOTX,DOTY,NR,HDFIX,PITFIX,VELFIX
7 ,ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4
8 ,XS,XP,P,KC
N = NUM3
FLAG2=0•0

```

```

505 DO 18 K = 1, NUMTRK
      PRINT 203,RG(K),AZG(K),ELG(K)
      END

```

```

203 FORMAT(//,30X,5HRC = ,F10.6,5X,6HAZG = ,F10.6,5X, 6HELG = ,F10.6)
      IF(ABSF(RADRNG(N)-PRADRNG(K))-RG(K)) 15,15,18
15   IF(AZRAD(N)-(360.0-AZG(K)) ) 302,303,303
302  IF(AZRAD(N)-AZG(K)) 300,300,304
300  IF(PAZRAD(K)-(360.0-AZG(K)) ) 304,3 1,301
301 AZY =AZRAD(N)+ 360.0
      IF(ABSF(AZY-PAZRAD(K))-AZG(K)) 16,16,18
304  IF(ABSF(AZRAD(N)-PAZRAD(K))-AZG(K)) 16,16,18
303  IF(PAZRAD(K)-AZG(K)) 305,304,304
305 PAZY=PAZRAD(K)+360.0
      IF(ABSF(PAZY-AZRAD(N))-AZG(K)) 16,16,18
16   IF(ABSF(ELRAD(N)-PELRAD(K))-ELG(K)) 17,17,18
17   IC=N
      J=K
      EC((IC,J)=1
      PRINT 60,((((IC,J),J=1,10),IC=1,10)
      FLAG2=1.0
      AZ4(N)=2.0
18   CONTINUE
      IF(FLAG2) 120,120,121
120  PRINT 536,N,NUMTRK
536  FORMAT(/,10X,11HTARGET NO. ,12,2X, 23HDID NOT CORRELATE WITH ,12,
1   6HTRACKS)
      60  FORMAT(//10X,10F10.3)
      IF(AZ4(N)=AZ4(N)) 62,62,63
63   AZ4(N)=AZ4(N) - 1.0
      GO TO 121
      62  NUMTRK= NUMTRK + 1
      KC(NUMTRK) = 0
      IF(NUMTRK =9) 123,538,538
538  NUMTRK= 9
123  C(N,NUMTRK) =1.0
      121 RETURN
      END
      SUBROUTINE ASSOC
      DIMENSION RANGE(10),TARGRNG( 10 ),PD(10),

```

```

11 TITLE(12), HD(10), VEL(10), ALT(10), PITCH(10), XT( 10 ), YT( 10 ),
22 ZT(10), AZT(10), C(10,10), S(10), ELT(10), XRAD(10), YRAD(10), ZRAD(10),
33 ELRAD(10), AZRAD(10), BOXMIN(10), BOXMAX(10), ELTEST(10), IAZ(10),
44 PRADRN(10), PELRAD(10), PAZRAD(10), RADRNG(10), XT1(200), YT1(200),
55 XRAD1(200), YRAD1(200), INDEX(10), INDE(10), CONT(10), IAT(10), PXRAD
66 (10), PYRAD(10), PZRAD(10), DOTX(10), DOTZ(10), DOTY(10), NR(10), HDFIX(
77 10), PITFIX(10), VELFIX(10), RG(10), AZG(10), ELG(10), R1(20), R2(20),
88 R3(20), R4(20), AZ1(20), AZ2(20), AZ3(20), AZ4(20),
DIMENSION XS(6,6), XP(6,6), P(6,6), KC(10)
COMMON  RANGE, TARGRNG, PDET, PD, IR, RANDOM, NUNIF, DEV,
11 RANERR, RADRNG, HD, VEL, ALT, PITCH, XT, YT, NT, JA, T, J, TEST,
22 AZT, ELT, IAZ, BOXMAX, BOXMIN, ELTEST, TEST, AZRAD1,
33 RADRNG1, ELRAD1, AZRAD, C, S, DOTT, ZT, PAZRAD, ELBEAMT
44 FLAG, NUMTRK, NUM1, NUM2, NUM3, AZBEAMT, RADDEG, ELRAD
55 , XRAD, YRAD, ZRAD, CONT, PRADRN, PELRAD, XT1, YT1, XRAD1, YRAD1, INDEX, IN
66 DE, IAT, PXRAD, PYRAD, PZRAD, DOTX, DOTZ, DOTY, NR, HDFIX, PITFIX, VELFIX
77 , ALPHA, BETA, RG, AZG, ELG, R1, R2, R3, R4, AZ1, AZ2, AZ3, AZ4
88 , XS, XP, P, KC
121 DO 19 J= 1,10
      SUM1=0
      DO 25 IC=1,10
25      SUM1=C(IC,J)+SUM1
         IF(SUM1-1.0)19,19,26
26      DO 119 IC=1,10
         IF(C(IC,J)-1.0)19,20,19
20      JP1=J+1
      DO 21 JP2=JP1,10
         IF(C(IC,JP2)-1.0)21,22,21
22      IP1=1
         SUM2=0
      DO 23 IP1=1,10
23      SUM2=C(IP1,JP2)+SUM2
         IF(SUM2-1.0)21,24,21
24      E(IC,J)=0
21      CONTINUE
119     CONTINUE

```

```

19 CONTINUE
DO 31 J=1,10
SUM3=0
DO 32 IC=1,10
SUM3=C(IC,J)+SUM3
IF (SUM3-1.) 31,31,33
33 DO 131 IC=1,10
IF (C(IC,J)) 31,31,36
36 SUM4=0
JP3=J+1
DO 35 JP4=JP3,10
35 SUM4=SUM4+C(IC,JP4)
IF (SUM4) 31,37,31
37 DO 231 IC=1,10
IF (C(IC,J)) 31,31,38
38 SUM5=0
JP5=J+1
DO 39 JP6=JP5,10
39 SUM5=SUM5+C(IC,JP6)
IF (SUM5) 31,31,40
40 C(IC,J)=0
231 CONTINUE
131 CONTINUE
31 CONTINUE
DO 41 J=1,10
SUM6=0
DO 42 IC=1,10
SUM6=SUM6+C(IC,J)
IF (SUM6-1.) 41,41,43
43 DO 44 IC=1,10
S(IC)=.210
IF (C(IC,J)) 44,44,45
45 S(IC)= PRADRNG(J) - RADRNG(IC)
44 CONTINUE
IC=1
IA=I+1

```

```

NB=1
S(NB)=•210
DO 46 IB=IA,10
IF (S(IC)-S(IB)) 46,46,47
47 IF (S(NB)-S(IB)) 46,46,48
48 NB=IB
46 CONTINUE
DO 49 IC=1,10
49 E(IC,J)=0
C(NB,J)=1
41 CONTINUE
DO 51 IC=1,10
SUM7=0
DO 52 J=1,10
52 SUM7=SUM7+C(IC,J)
IF (SUM7-1.) 51,51,53
53 DO 54 J=1,10
54 S(J)=•210
IF (C(IC,J)) 54,54,555
555 S(J) = PRADRNG(J) - RADRNG(IC)
54 CONTINUE
J=1
JZ = J +1
NC=1
S(NC)=101•0
DO 56 JB = JZ,10
IF (S(J)-S(JB)) 56,56,57
57 IF (S(NC)-S(JB)) 56,56,58
58 NC=JB
56 CONTINUE
DO 59 J=1,10
59 C(IC,J)=0
C(IC,NC) = 1
51 CONTINUE
PRINT 63
63 FORMAT (/10X,24H CORRELATION MATRIX)

```

```

PRINT 60,((C(IC,J),J=1,10),IC=1,10)
60 FORMAT(//10X,10F10.3)
END

```

```

SUBROUTINE TARGEN
  DIMENSION  RANGE(10),TARGRNG( 10 ),PD(10),
  1  TTITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10 ),YT( 10 ),
  2  ZT(10),AZT(10),C(10,10)*S(10),ELT(10)*XRAD(10)*YRAD(10)*ZRAD(10),
  3  ELRAD(10)*AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
  4  PRADRNG(10)*PELRAD(10)*PAZRAD(10),RADRNG(10),XT1(200)*YT1(200),
  5  XRAD1(200)*YRAD1(200),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
  6 (10),PYRAD(10)*PZRAD(10),DOTX(10),DOTZ(10),DOTY(10)*NR(10),HDFIX(
  7 10),PITFIX(10),VELFIX(10)*RG(10)*AZG(10)*ELG(10),R1(20),R2(20),
  8  R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
  DIMENSION XS(6,6),XP(6,6),P(6,6),KC(10)
  COMMON  RANGE, TARGRNG, PDET, PD, IR, RANDOM, NUNIF, DEV,
  1  RANERR, RADRNG, HD, VEL, ALT, PITCH, XT, YT, NT, JA, T, I, TEST,
  2  AZT, ELT, IAZ, BOXMAX, BOXMIN, ELTEST, TESTT, AZRAD1,
  3  RADRNG1,ELRAD1, AZRAD, C, S, DOTT, ZT, PAZRAD, ELBEAMT
  4,FLAG, NUMTRK, NUM1, NUM2, NUM3, AZBEAMT, RADDEG, ELRAD
  5, XRAD, YRAD, ZRAD, CONT, PRADRNG, PELRAD, XT1, YT1, XRAD1, YRAD1, INDEX, IN
  6 DE, IAT, PXRAD, PYRAD, PZRAD, DOTX, DOTZ, DOTY, NR, HDFIX, PITFIX, VELFIX
  7 ,ALPHA, BETA, RG, AZG, ELG, R1, R2, R3, R4, AZ1, AZ2, AZ3, AZ4
  8 *XS,XP,P,KC
  DO 1 K=1,NT
  14 IF(HD(K)-360.0) 24,24,23
  23 HD(K)=HD(K)-360.0
  24 THETA=(-360.0-HD(K) +90.0)/RADDEG
  16 PITCH1=PITCH(K)/57.2957795131
  DELTA=(VEL(K)*T )/3600.
  XT(K)=XT(K) +(COSF(THETA))*DELTA
  YT(K)=YT(K) +(SINF(THETA))*DELTA
  ZT(K)=ZT(K) +(SINF(PITCH1))* DELTA
  AZ1= ATANF( YT(K) /XT(K ))* 57.2957795131
  17 IF( XT(K) 11,7,7
  7 AZT(K) = 90.-AZ1
  GO TO 8

```

```

11 AZT(K) = 270.0 -AZ1
  8 TARGRNG(K) = SQRTF((XT(K))**2 + (YT(K))**2 +
1 (ZT(K))**2)
    ELT(K) = ATANF(ZT(K) / SQRTF((XT(K))**2 +
1 +(YT(K))**2)) *RADDEG
    ALT(K)=60.76.1*ZT(K)
1 CONTINUE
    RETURN
END

SUBROUTINE SETUP
DIMENSION RANGE(10),TARGRNG(10),PD(10),
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT(10),YT(10),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),YRAD(10),ZRAD(10),
3 ELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRNG(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(200),YT1(200),
5 XRAD1(200),YRAD1(200),INDEX(10),INDE(10),IAT(10),PXRAD
6 (10),PYRAD(10),PZRAD(10),DOTX(10),DOTZ(10),DOTH(10),NR(10),HDFIX(
7 10),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8 R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
DIMENSION HPHT(6,6),X(6,6),VN1(6,6),Z(6,6),PHT(6,6),
1 PHIT(6,6),P1(6,6),H(6,6),HT(6,6),PHI(6,6),R(6,6),Q(6,6),P(6,6),
2 XS(6,6),XP(6,6),XP1(6,6),T1(6,6),T2(6,6),Q1(6,6),Q8(6,6),PHI1(6,6)
3,PHI8(6,6)

DIMENSION RDOT(10),AZDOT(10),ELDOT(10),
1 KC(10),PK1(6,6),PK2(6,6),PK3(6,6),PK4(6,6),PK5(6,6),PK6(6,6),
2 PK7(6,6),PK8(6,6),PK9(6,6)
COMMON RANGE,TARGRNG,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TESTT,AZRAD1,
3 RADRNG1,ELRAD1, AZRAD ,C,S,DOTT,ZT,PAZRAD,ELBEAMT
4,FLAG, NUMTRK, NUM1 ,NUM2,NUM3,AZBEAMT,RADDEG,ELRAD
5 ,XRAD,YRAD,ZRAD,CONT,PRADRNG,PELRAD,XT1,YT1,XRAD1,YRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DOTH,NR,HDFIX,PITFIX,VELFIX
7 ,ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4
8,XS,XP,P,KC
IF(I-1) 425,425,61

```

```

425 DO 104 J=1,10
104 KC(J)=0
DO 488 L=1,6
DO 488 M=1,6
PK1(L,M)=P(L,M)
PK2(L,M)=P(L,M)
PK4(L,M)=P(L,M)
PK5(L,M)=P(L,M)
PK6(L,M)=P(L,M)
PK7(L,M)=P(L,M)
PK8(L,M)=P(L,M)
PK9(L,M)=P(L,M)
PK3(L,M)=P(L,M)
488 DO 777 L=1,NUMTRK
61 DO 777 L=1,NUMTRK
23 NUM2 = T
DO 21 IC= 1,10
IF(C(IC,L)) 21,21,22
21 CONTINUE
XP(1,1)= PRADRNG(L)
XP(2,1)= RDOT(L)
XP(3,1)= PAZRAD(L)
XP(4,1)= AZDOT(L)
XP(5,1)= PELRAD(L)
XP(6,1)= ELDOT(L)
DO 296 M=1,6
DO 296 N=1,6
296 P(M,N)=0.0
P(1,1)=0.09
P(2,2)= 0.0278
P(3,3)= 2.0
P(4,4)=0.25
P(5,5)=0.25
P(6,6)= 0.00109
Z(1,1)= PRADRNG(L)
Z(2,1)= PAZRAD(L)
Z(3,1)= PELRAD(L)

```

```

GO TO 267
22 LT=L
      IF(KC(LT)) 264,264,265
264  XP(1,1)= RADRNG(IC)
      XP(2,1)= -0.166
      XP(3,1)= AZRAD(IC)
      XP(4,1)= 0.0
      XP(5,1)= ELRAD(IC)
      XP(6,1)=0.0
DO 266 M=1,6
DO 266 N=1,6
266  P(M,N)=0.0
      P(1,1)=0.09
      P(2,2)= 0.0278
      P(3,3)= 2.0
      P(4,4)=0.25
      P(5,5)=0.25
      P(6,6)= 0.00109
      Z(1,1)=RADRNG(IC)
      Z(2,1)= AZRAD(IC)
      Z(3,1)= ELRAD(IC)
      KC(LT)= LT
GO TO 267
265  XP(1,1)= PRADRNG(L)
      XP(2,1)= RDOT(L)
      XP(3,1)= PAZRAD(L)
      XP(4,1)= AZDOT(L)
      XP(5,1)= PELRAD(L)
      XP(6,1)= ELDOT(L)
      Z(1,1)= RADRNG(IC)
      Z(2,1)= AZRAD(IC)
      Z(3,1)= ELRAD(IC)
      IF(L-2) 268,269,270
268  DO 271 J=1,6
      DO 271 JJ=1,6
271  P(J,JJ)=PK1(J,JJ)

```

```

60 TO 267
269 DO 272 J=1,6
DO 272 JJ=1,6
272 P(J,JJ)=PK2(J,J)
60 TO 267
270 IF(L-3) 288,288,289
288 DO 273 J=1,6
DO 273 JJ=1,6
273 P(J,JJ)=PK3(J,J)
60 TO 267
289 IF(L-5) 274,275,276
274 DO 277 J=1,6
DO 277 JJ=1,6
277 P(J,JJ)=PK4(J,J)
60 TO 267
275 DO 278 J=1,6
DO 278 JJ=1,6
278 P(J,JJ)=PK5(J,J)
60 TO 267
276 IF(L-6) 290,290,291
290 DO 279 J=1,6
DO 279 JJ=1,6
279 P(J,JJ)=PK6(J,J)
60 TO 267
291 IF(L-8) 280,281,282
280 DO 283 J=1,6
DO 283 JJ=1,6
283 P(J,JJ)=PK7(J,J)
60 TO 267
281 DO 284 J=1,6
DO 284 JJ=1,6
284 P(J,JJ)=PK8(J,J)
60 TO 267
282 DO 285 J=1,6
DO 285 JJ=1,6
285 P(J,JJ)=PK9(J,J)

```

```

267 CALL KALFILT(Z,XS,XP,P,NUM1,NUM2)
      IF (IC (IC,L)) 263,263,400
      400 RADRNG (IC) = XS(1,1)
      AZRAD (IC) = XS(3,1)
      ELRAD (IC) = XS(5,1)
263   RDOT (L) = XP(2,1)
      AZDOT (L) = XP(4,1)
      ELDOT (L) = XP(6,1)
      PRADRNG (L) = XP(1,1)
      PAZRAD (L) = XP(3,1)
      PELRAD (L) = XP(5,1)
      RG (L) = 2.0 * SQRTF(P(1,1))
      AZG (L) = 1.0 * SQRTF(P(3,3))
      ELC (L) = 2.5 * SQRTF(P(5,5))
      IF (L-2) 401,402,403
401   DO 404 J=1,6
      DO 404 JJ=1,6
404   PK1 (J,JJ) = P (J,JJ)
      GO TO 777
402   DO 405 J=1,6
      DO 405 JJ=1,6
405   PK2 (J,JJ) = P (J,JJ)
      GO TO 777
403   IF (L-4) 406,407,408
406   DO 409 J=1,6
      DO 409 JJ=1,6
409   PK3 (J,JJ) = P (J,JJ)
      GO TO 777
407   DO 410 J=1,6
      DO 410 JJ=1,6
410   PK4 (J,JJ) = P (J,JJ)
      GO TO 777
408   IF (L-6) 411,412,413
411   DO 414 J=1,6
      DO 414 JJ=1,6
414   PK5 (J,JJ) = P (J,JJ)

```

```

GO TO 777
412 DO 415 J=1,6
DO 415 JJ=1,6
415 PK6(J,JJ)=P(J,JJ)
GO TO 777
413 IF(L-8) 416,417,418
416 DO 419 J=1,6
DO 419 JJ=1,6
417 PK7(J,JJ)=P(J,JJ)
GO TO 777
417 DO 420 J=1,6
DO 420 JJ=1,6
420 PK8(J,JJ)=P(J,JJ)
GO TO 777
418 DO 421 J=1,6
DO 421 JJ=1,6
421 PK9(J,JJ)=P(J,JJ)
777 CONTINUE
END
SUBROUTINE TWS
DIMENSION
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT(10),YT(10),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),ZRAD(10),
3 ELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRN(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(200),YT1(200),
5 XRAD1(200),YRAD1(200),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
6(10),PYRAD(10),PZRAD(10),DOTTX(10),DOTTZ(10),DOTY(10),NR(10),HDFIX(
710),PITFFIX(10),VELFFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
9,XT2(30),YT2(30),XRAD2(30),YRAD2(3)
DIMENSIONXS(6,6),XP(6,6),P(6,6),KC(10)
COMMON
1 RANERR, RANGE,TARGRNG,PDET,PD,IR, RANDOM, NUNIF, DEV,
2 AZT , ELT , IAZ , BOXMAX, BOXMIN, ELTEST, TESTT, AZRAD1,
3 RADRNG1,ELRAD1, AZRAD ,CS,DOTT,ZT,PAZRAD,ELBEAMT

```

```

4,FLAG, NUMTRK, NUM1, NUM2, NUM3, AZBEAMT, RADDEG, ELRAD
5, XRAD, YRAD, ZRAD, CONT, PRADRN, PELRAD, XT1, YT1, XRAD1, YRAD1, INDEX, IN
6 DE, IAT, PXRAD, PYRAD, PZRAD, DOTX, DOTY, NR, HDFIX, PITFIX, YELFIX
7 ALPHA, BETA, RG, AZG, ELG, R1, R2, R3, R4, AZ1, AZ2, AZ3, AZ4
8, XS, XP, P, KC
ITITLE(1)=8HSECT SCA
ITITLE(2)=8HN MANEUV
ITITLE(3)=8HING TARG
ITITLE(4)=8HO DEG/SE
ITITLE(5)=8HC
ITITLE(6)=8HKALMAN
ITITLE(7)=8H DELANEY
ITITLE(8)=8H W.F.
ITITLE(9)=8H
ITITLE(10)=8HJOB0194
ITITLE(11)=8HTEST RUN
ITITLE(12)=8H 2
IXX = 20
I= 1
T= .5
DO 105 LL=1,200
XT1(LL)=0.0
YT1(LL)=0.0
XRAD1(LL)=0.0
YRAD1(LL)=0.0
105 CONTINUE
LL=
EB=
NUMB=0
NUMTAR=1
K=NUM3
NR(K)=K
IAZREF=AZRAD(K)
PRINT 60,NT,K,IAZREF
60 FORMAT( /,10X,3HNT=,12,5X,2HK=,12,5X,7HIAZREF=,14)
LIML=IAZREF-44

```

```

1 IF (LIML) 2,4,4
2 LIML=360+LIML
4 LIMR=IAZREF +44
5 IF (LIMR-360) 6,6,5
6 LIMR=LIMR-360
7 LIMB =1
8 LIMT=29
9 ITEL1=LIMB
10 ITEMP1 = LIML
11 NUMB=NUMB+1
12 IF ( NUMB - 30 ) 43,43,44
13 PRINT 111, NUMB
14 FORMAT( /,10X,7HNUMB = , 12)
15 IF (I-2)160,158,158
16 DO 150 J= 1,NUMTRK
17 TOTAL = 0.0
18 DO 151 IC = 1,10
19 TOTAL =TOTAL+ C(IC,J)
20 IF (TOTAL) 152,150
21 IF (INDEX(J)) 156,156,153
22 IF (I-(INDEX(J) +1)) 156,154,156
23 INDEX(J) =1
24 INDEX(J) = 1
25 GO TO 150
26 DO 155 JB=J,NUMTRK
27 DO 157 IC= 1,10
28 C(IC,JB )= C(IC,JB+1)
29 PRADRNG(JB)=PRADRNG(JB+1)
30 PAZRAD(JB)=PAZRAD(JB+1)
31 INDEX(JB)=INDEX(JB+1)
32 INDE(JB)=INDE(JB+1)
33 PELRAD(JB)=PELRAD(JB+1)
34 155 CONTINUE
35 PRINT 121,J
36 121 FORMAT( //,10X,11HDROP TRACK ,12)
37 NUMTRK =NUMTRK -1

```

```

150 CONTINUE
  DO 55 IC=1,10
  DO 55 J= 1,10
  55 C(IC,J)=0.0
  160 DO 19 J=1,NT
      PRINT 61,J, NR(J)
  61  FORMAT( /,10X,2HJ=,I2,6HNR(J)=,I2 )
      IF( NR(J) 19,19,23
  23  ITEL=ELRAD(NR(J))
      TEL=ITEL
      ITEMp = AZRAD(NR(J))
      TEMP=ITEMp
      IF(TEL) 75,70,75
  75  IF( (ELRAD(NR(J)) -TEL) -0.5000) 77,70,70
  70  TEL =TEL+1.0
  77  IF(TEL-29.0) 71,71,76
  76  TEL=TEL- 1.0
  71  IF( (AZRAD(NR(J))-TEMP)-0.5000) 73,72,72
  72  TEMP =TEMP+1.0
  73  PRINT 74, TEL,ELRAD(NR(J)),TEMP,AZRAD(NR(J))
  74  FORMAT( /,10X,7H TEL = ,F10.4,5X,7H ELRAD= ,F10.4, 5X,
  1 7HTEMP = ,F10.4,5X,8HAZRAD = ,F10.4)
      C
      C   THIS SECTION LOOKS AT KNOWN TARGETS
      C
  DO 18 K=1,NT
      IF(ABSF( TEL - ELT(K))- 1.0) 16,16,18
  16  IF( AZT(K) -1.0) 92,92,94
  92  IF( 359.0 -TEMP) 93,93,94
  93  AZTT=AZT(K) + 360.0
      IF( ABSF(TEMP -AZTT) -1.0) 17,18,18
  94  IF(ABSF( TEMP - AZT(K) ) -1.0) 17,18,18
  17  NUM3=K
      PRINT 62,TEL,ELT(K),TEMP,AZT(K),K
  62  FORMAT( /,10X,4HTEL=,F10.4,5X,9H ELT(K)=,F10.4,5X,5HTEMP=,F10.4,5X,
      17HAZT(K)=,F10.4,5X,2HK=,I2)

```

```

CALL DETECT
IF (RADRNG(K)) 18,18,33
18 CALL RNDEV ( NUNIF ,DEV)
AZRAD(K)=AZT(K)+DEV*0.05
CALL RNDEV(NUNIF,DEV)
ELRAD(K)=ELT(K)+DEV*0.05
PRINT 20,K,RADRNG(K),AZRAD(K),ELRAD(K)
PRINT 20,K,RADRNG(K),AZRAD(K),ELRAD(K)
20 FORMAT(//,10X,7HTARGET ,12.5X, 9HRADRNG = ,F10.4,5X,8HAZRAD = ,
1F10.4, 5X, 8HELRAD = , F10.4)
PRINT 50
50 FORMAT(//,10X, 35HCALLING CORRASS AFTER STATEMENT 20 )
CALL CORRASS
18 GOTO 19
19 CONTINUE
C
C THIS SECTION SCANS THE SECTOR ASTIME REMAINING PERMITS.
C IT IS DIVIDED INTO 0.5 SECONDS OF KNOWN TARGETS PLUS SCAN
C AND 0.5SECONDS OF SCAN
C
C
FLAG3= 0.0
INC= 100 -(NUMTAR *5)
102 IF( FLAG3 -1.0) 100,101,101
101 INC=(200 - NUMTAR *5)- INC
100 PRINT 81, INC
81 FORMAT(/,10X,6HINC = , I4)
DO 80 J=1,INC
TEL=ITEL1
TEMP=ITEMP1
37 DO 26 K=1,NT
IF (ABSF(TEL - ELT(K))-1.0) 25,25,26
25 IF (ABSF( TEMP -AZT(K)) -1.0) 45,26,26
45 NUM3=K
CALL DETECT
IF (RADRNG(K)) 26,26,34
34 CALL RNDEV (NUNIF,DEV)
AZRAD(K) = AZT(K) +DEV/100.

```

```

CALL RNDEV (NUNIF,DEV)
ELRAD(K) = ELT(K)+DEV/100.
PRINT 20,K,RADRNG(K),AZRAD(K),ELRAD(K)
CALL CORRASS
106  NR(K)=NUMTAR+1
NR(K)=K
26  CONTINUE
ITEMP1=ITEMP1+2
IF(ITEMP1-360) 91,91,90
90  ITEMPI=1
91  IF( LIMR -90) 82,83,83
82  IF((ITEMP1 +90)-360) 83,80,80
83  TEMP=ITEMP1
ITEMP = TEMP
IF( LIMR-ITEMP) 84,80,80
84  ITEMPI=LIML
ITEMI1=ITEMI1+2
IF( ITEMI1- 29) 80,80,85
85  ITEMI1=1
80  CONTINUE
36  CALL TARGEN
CALL ASSOC
CALL SETUP
DO 56 K=1,NT
XRAD(K) = (RADRNG(K)*COSF( ELRAD(K )/57.29577) ) *
1 SIN( AZRAD(K ) / 57.29577 )
YRAD(K) = (RADRNG(K)*COSF( ELRAD(K )/57.29577) ) *
1 COSF( AZRAD(K ) / 57.29577 )
LL = NUMB
IF( (K-1)200,200,201
200  XT1(LL)= XT(K)
YT1(LL)= YT(K)
58  CC= .0
DO 51 J=1,10
51  CC=CC+C(K,J)

```

```

1 IF(CC) 54,54,52
52 XRADI(LL)=XRAD(K)
  YRAD1(LL)=YRAD(K)
  GO TO 56
54 XRADI(LL)=0.0
  YRAD1(LL)=0.0
  GO TO 56
201 XT2(LL)=XT(K)
  YT2(LL)=YT(K)
558 CC= 0
  DO 551 J=1,10
551 CC=CC+C(K,J)
  IF(CC) 554,554,552
552 XRAD2(LL)=XRAD(K)
  YRAD2(LL)=YRAD(K)
  GO TO 56
554 XRAD2(LL)= 0.0
  YRAD2(LL)= 0.0
56 CONTINUE
  IF(LL - IXX) 301,300,301
301 FLAG3=FLAG3+ 1.0
  FLAG3= FLAG3+1.0
  IF(FLAG3 -2.0) 102,28,28
102 CALL DRAW(LL,XT1,YT1,1,0,LABEL,ITITLE, 1.,0,0,2,2,5, 8,0, LAST)
  CALL DRAW(LL,XT2,YT2,2,0,LABEL,ITITLE, 1.,0,0,2,2,5, 8,0, LAST)
  CALL DRAW(LL,XRAD2,2,5,LABEL,ITITLE, 1.,1,0,3,2,2,6,10,0,
1LAST)
  CALL DRAW(LL,XRAD1,YRAD1,3,1,LABEL,ITITLE, 3.,3,0,3,2,2,6,10,0,
1LAST)
  IXX = 21
  IF( NUMB - 21) 302,44,44
302 NUMB= 0
  GO TO 301
44 RETURN
END
SUBROUTINE KALFILT(Z,XS,XP,P,NUM1,NUM2)

```

```

DIMENSION HPHT(6,6),X(6,6),VNI(6,6),Z(6,6),PHT(6,6),
1PHIT(6,6), P1(6,6),H(6,6),HT(6,6),PHI(6,6),R(6,6),Q(6,6),P(6,6),
2XS(6,6),XP(6,6),XP1(6,6),T1(6,6),T2(6,6),Q1(6,6),Q8(6,6),PHI1(6,6),
3,PHI8(6,6)

C THE FIRST SECTION OF KALFILT SETS UP THE INITIAL MATRICES REQUIRED
C IN THE RECURSION EQUATIONS
C SECTION ONE IS EXECUTED IF NUM1=0
C NUM3 = K OR TARGET NUMBER
C NUM2 = T I.E. 1 OR 8 SEC. ACTS AS A FLAG TO PICK THE CORRESPONDING Q
C AND PHI MATRIX

C IF (NUM1=1) 61,60,60
61 PRINT 1
1 FORMAT(//,40X,26H OBSERVABILITY OR H MATRIX )
CALL READD(3,6,H)
PRINT 2
2 FORMAT(//,40X,19H H TRANSPOSE OR HT )
CALL TRANS(H,3,6,HT)
DO 3 I=1,6
3 PRINT 4,(HT(I,J),J=1,3)
4 FORMAT(/,3F13.5)
PRINT 5
5 FORMAT(//,20X,43HCOVARIANCE OF MEASUREMENT NOISE OR R MATRIX )
CALL READD(3,3,R)
PRINT 6
6 FORMAT(//,40X,28HSTATE TRANSITION OR PHI MATRIX )
CALL READD(6,6,PHI1)
PRINT 6
CALL READD(6,6,PHI8)
PRINT 7
7 FORMAT(//,20X,39HCOVARIANCE OF PERTURBATIONS OR Q MATRIX )
CALL READD(6,6,Q1)
PRINT 7
CALL READD(6,6,Q8)

```

```

C
60 IF( NUM2 -1) 62,62,64
62 DO 63 L=1,6
DO 63 M=1,6
63 Q(L,M)=Q1(L,M)
63 PHI(L,M)=PHI1(L,M)
64 GO TO 66
64 DO 65 L=1,6
DO 65 M= 1,6
65 Q(L,M)=Q8(L,M)
65 PHI(L,M)=PHI8(L,M)
66 NUM1= 1

C Z IS THE OBSERVABILITY VECTOR TIMES X PLUS NOISE
C
C CALCULATE THE GAIN MATRIX G(N)
C
CALL PROD(P,HT,6,6,3,PHT)
CALL PROD(H,PHT,3,6,3,HPHT)
CALL ADD(HPHT,R,3,3,HPHT)
CALL RECIP(3,0.000001,HPHT,VNI,KER)
IF(KER-2) 92,91,92
91 PRINT 90, KER
90 FORMAT(//,10X,6HKER = ,I2)
92 CALL PROD(PHT,VNI,6,3,3,G)

C CALCULATE THE SMOOTH X VECTOR OR XS
C
PRINT 100
100 FORMAT(//,20X,2HXP)
CALL PRINTT(S,1,XP)
PRINT 101
101 FORMAT(//,20X,1HZ)
CALL PRINTT(3,1,Z)
CALL PROD(H,XP, 3,6,1,XP1)

```

```

DO 10 I=1,3
10  XP1(I,1) = - XP1(I,1)
CALL ADD(Z,XP1,3,1,Z)
DO 31 I=1,6
DO 31 J=1,6
31  T1(I,J) =0.0
CALL PROD(G,Z,6,3,1,T1)
CALL ADD(XP,T1,6,1,XS)
PRINT 102
102 FORMAT(//,20X,2HXS)
CALL PRINTT(6,1,XS)

C CALCULATE THE COVARIANCE OF THE ERROR MATRIX PNN
C
CALL PROD(H,P,3,6,6,P1)
DO 32 I=1,6
DO 32 J= 1,6
32  T2(I,J) = 0.0
CALL PROD(G,P1,6,3,6,T2)
DO 11 I=1,6
DO 11 J=1,6
11  T2(I,J) = - T2(I,J)
CALL ADD(P,T2,6,6,P)

C CALCULATE THE PRIDICTED VALUE OF THE X VECTOR XN/N-1
C
CALL PROD(PHI,XS,6,6,1,XP)
PRINT 100
CALL PRINTT(6,1,XP)

C CALCULATE THE PREDICTED COVARIANCE MATRIX OF ERROR PN/N-1
C
CALL TRANS(PHI,6,6,PHIT)
PRINT 14
14 FORMAT(//,40X,20HPHI TRANSPOSE MATRIX )
CALL PRINTT(6,6,PHIT)

```

```

CALL PROD(P,PHIT,6,6,6,P1)
DO 30 I=1,6
DO 30 J=1,6
30 T1(I,J) = 0.0
    CALL PROD(PHI,P1,6,6,6,T1)
    CALL ADD(T1,Q,6,6,P)
    PRINT 103
103 FORMAT(//,20X,6HPN/N-1)
    CALL PRINTT(6,6,P)
85 RETURN
END

SUBROUTINE READD (M,N,XX)
DIMENSION XX(6,6)
DO 1 I=1,M
READ 2 ,(XX(I,J), J=1,N)
2 FORMAT( 6F13.5)
1 PRINT 2,(XX(I,J), J=1,N )
END

SUBROUTINE PRINTT (M,N,XX)
DIMENSION XX(6,6)
DO 1 I=1,M
PRINT 2,(XX(I,J), J=1,N )
2 FORMAT( /, 6F13.5)
END

SUBROUTINE TRANS(A,N,M,D)
DIMENSION A(6,6),D(6,6)
DO 153 I=1,N
DO 153 J=1,M
153 D(J,I)=A(I,J)
END

SUBROUTINE PROD(A,B,N,M,L,D)
DIMENSION A(6,6),B(6,6),D(6,6)
DO 151 I=1,N
DO 151 J=1,L
D(I,J)=0.0
DO 151 K=1,M

```

```

151 D(I,J)=D(I,J)+A(I,K)*B(K,J)
END
SUBROUTINE ADD(A,B,N,M,D)
DIMENSION A(6,6),B(6,6),D(6,6)
DO 152 I=1,N
DO 152 J=1,M
152 D(I,J)=A(I,J)+B(I,J)
END
SUBROUTINE RECIP(N,EP,B,X,KER)
DIMENSION A(6,6),B(5,6),X(6,6)
DO 1 I=1,N
DO 1 J=1,N
A(I,J)=B(I,J)
1 X(I,J)=0.0
1 DO 2 K=1,N
2 X(K,K)=1.0
10 DO 34 L=1,N
34 KP=9
Z=0.0
DO 12 K=L,N
12 IF (Z-ABSF(A(K,L))) 11,12,12
11 Z=ABSF(A(K,L))
11 KP=K
12 CONTINUE
13 DO 14 J=L,N
13 Z=A(L,J)
13 A(L,J)=A(KP,J)
14 A(KP,J)=Z
14 DO 15 J=1,N
15 Z=X(L,J)
15 X(L,J)=X(KP,J)
15 X(KP,J)=Z
20 IF (ABSF(A(L,L))-EP) 50,50,30
30 IF (L-N) 31,34,34
31 LPL=L+1
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027

```

```

0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048

DO 36 K=LP1,N
  IF (A(K,L))32,36,32
32  RATIO=A(K,L)/A(L,L)
  DO 33 J=LP1,N
33  A(K,J)=A(K,J)-RATIO*A(L,J)
  DO 35 J=1,N
35  X(K,J)=X(K,J)-RATIO*X(L,J)
36  CONTINUE
34  CONTINUE
40  DO 43 I=1,N
  I1=N+1-I
  DO 43 J=1,N
  S=0.0
  IF (I1-N)41,43,43
41  I1P1=I1+1
  DO 42 K=I1P1,N
42  S=S A(I1,K)*X(K,J)
43  X(I1,J)=(X(I1,J)-S)/A(I1,I1)
  KER=1
  RETURN
50  KER=2
  RETURN
END

```

PROGRAM KALMON1

```

DIMENSION RANGE(10),TARGRNG( 10 ),PD(10),
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10 ),YT( 10 ),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),YRAD(10),ZRAD(10),
3 ZELRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRNG(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(100),YT1(100),
5 XRAD1(100),YRAD1(100),INDEX(10),INDE(10),CONT(10),IAT(10),PXRAD
6(10),PYRAD(10),PZRAD(10),DOTX(10),DOTZ(10),DOTY(10),NR(10),HDFIX(
710),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(40),R2(40),
8R3(40),R4(40),AZ1(40),AZ2(40),AZ3(40),AZ4(40),VMAG1(40),VMAG2(40),
9VMAG3(40),VMAG4(40),VPHI1(40),VPHI2(40),VPHI3(40),VPHI4(40)
DIMENSION A(6,6),B(6,6),D(6,6),X(6,6),
1 PHIT(6,6),P1(6,6),H(6,6),HT(6,6),PHI(6,6),R(6,6),Q(6,6),P(6,6),
2 XS(6,6),XP(6,6),XX(6,6),VNI(6,6),Z(6,6),PHT(6,6),HPHT(6,6),XP1(
3 6,6),T1(6,6),T2(6,6)
COMMON RANGE,TARGRNG,PDET,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,T,I,TEST,
2 AZT , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TESTT,AZRAD1,
3 ZRADRNG1,ELRAD1, AZRAD ,C,S,DOTT,ZT,PAZRAD,ELBEAMT
4,FLAG, NUMTRK, NUM1 ,NUM2,NUM3,AZBEAMT, RADDEG,ELRAD
5, XRAD, YRAD,ZRAD,CONT,PRADRNG,PELRAD,XT1,XRAD1,YRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DTY,NR,HDFIX,PITFIX,VELFIX
7 ,ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4,VMAG1,VMAG2,
8VMAG3,VMAG4,VPHI1,VPHI2,VPHI3,VPHI4
100 FORMAT(4 F10.4)
101 FORMAT(/,3X,11H HEADING = ,F10.4, 7HDEGREES, 3X,10HVELOCITY = ,
1F10.4, 8HN.M./HR.3X,11HALTITUDE = ,F10.4, 4HFEET,3X, 7HPITCH = ,
2F10.4, 7DEGREES )
102 FORMAT( 2F10.4)
C
C TEST RUN FOR MONTE CARLO AVERAGES
C THIS PROGRAM TESTS THE KALMAN FILTER FOR AN OBSERVATION TIME OF T=1.0
C VMAG2(L) = ENSEMBLE AVERAGE OF RADIFF
C VPHI2(L) = ENSEMBLE AVERAGE OF AZDIFF
C AZ2 (L) = ENSEMBLE AVERAGE OF ELDIFF

```

C NB = THE NUMBER OF RUNS
C NBT = NUMBER OF DISCRETE TIMES
NB=100
NBT=20
T=1.0
NT =1

C C

```
ITITLE (1)=8HHD160/V
ITITLE (2)=8HTEL600 /
ITITLE (3)=8HALT500/
ITITLE (4)=8HLEV/RT3
ITITLE (5)=8HDS/T=1
ITITLE (6)=8H
ITITLE (7)=8H
ITITLE (8)=8H
ITITLE (9)=8HJOB0194
ITITLE (10)=8HDELANEY
ITITLE (11)=8H W. F.
ITITLE (12)=8HKALMON1
RADDEG= 57.2957795131
NUNIF = 12207003125.
DO 714 K=1,NT
READ 100, HD(K), VEL(K), ALT(K), PITCH(K)
ZT(K)=ALT(K)/ 6076.1
714 READ 102, XT(K), YT(K)
HEAD=HD(1)
VELOC= VEL(1)
ALIT= ALT(1)
PIT= PITCH(1)
ZTT= ZT(1)
XPOS= XT(1)
YPOS= YT(1)
NUM2=0
DO 804 JK=1,NB
HD(1)= HEAD
```

```

VEL(1)= VELOC
ALT(1)= ALTTIT
PITCH(1)= PIT
ZT(1)= ZTT
XT(1)= XPOS
YT(1)= YPOS
CALL TARGEN
721 DO 708 I=1,NBT
K=1
XRAD1(I)= 1
CALL RNDEV(NUNIF,DEV)
RADERR=DEV*0.3
RADRNG(K)= TARGRNG(K)+ RADERR
CALL RNDEV(NUNIF,DEV)
AZERR= DEV*0.5
AZRAD(K)= AZT(K)+ AZERR
CALL RNDEV(NUNIF,DEV)
ELERR= DEV*0.5
ELRAD(K)=ELT(K)+ELERR
IF(ELRAD(K) .LT. 300 .OR. 301 .GT. 300)
ELRAD(K)= 0.0
301 NUMB=I
CALL TARGEN
K=1
168 RADIFF=ABSF(XP(1,1))-ABSF(TARGRNG(K))
AZDIFF=ABSF(XP(3,1))-ABSF(AZT(K))
IF(AZDIFF-10.0) 165,165,166
165 IF(10.0-AZDIFF) 166,265,265
166 AZDIFF=ABSF(360.0-AZDIFF)
265 ELDIFF=ABSF(XP(5,1))-ABSF(ELT(K))
R1(NUMB)=RADIFF**2
R4(NUMB)=AZDIFF**2
AZ4(NUMB)= ELDIFF**2
C AZ2 REPRESENTS THE X COMPONENT OF SMOOTH POSITION.
C AZ3 REPRESENTS THE Y COMPONENT OF SMOOTH POSITION.
C AZ2(NUMB)=(XS(1,1)*COSF(XS(5,1)/RADDEG))*SINF(XS(3,1)/RADDEG)

```

```

AZ3(NUMB)=(XS(1,1)*COSF(XS(5,1)/RADDEG))*COSF(XS(3,1)/RADDEG)
R2(NUMB)=XT(K)
R3(NUMB)=YT(K)
708 CONTINUE
NUM2=1
PRINT 671, JK
PRINT 671 FORMAT(//,20X,4HJK= ,12)
PRINT 667
667 FORMAT(//,30X,9HRADIFF**2)
PRINT 668, (R1(J), J=1,15)
668 FORMAT(//,1X,15F7.4)
PRINT 669
669 FORMAT(//,30X,9HAZDIFF**2)
PRINT 668, (R4(J), J=1,15)
PRINT 670
670 FORMAT(//,30X,9HELDIFF**2)
PRINT 668, (AZ4(J), J=1,15)
CALL MONTE (NB,NBT, JK,R1,R4,AZ4,NUMB)
IF (JK-NB) 804,999,804
999 LABEL=4HACTU
CALL PLOT (NUMB,R2,R3,0,0,LABEL,ITITLE,2.,2.,5,5,2,2,10,0,0, LAST)
LABEL=4HSMT
CALL PLOT (NUMB,AZ2,AZ3,0,0,LABEL,ITITLE,2.,2.,5,5,2,2,10,0,0, LAST
1 )
804 CONTINUE
END
116
SUBROUTINE MONTE (NB,NBT, JK,R1,R4,AZ4,NUMB)
DIMENSION R1(40),R4(40),AZ4(40),VMAG2(40),VPHI2(40),XRAD1(100)
1,ITITLE(12),AZ2(40)
IF ( JK-1) 811,811,812
811 DO 813 LL=1,NBT
VMAG2(LL)=0.0
VPHI2(LL)=0.0
813 AZ2(LL)=0.0
812 DO 806 L=1,NBT
VMAG2(LL)=VMAG2(LL)+R1(L)

```

```

VPHI12(L)= VPHI12(L) +R4(L)
AZ2(L)= AZ2(L) +AZ4(L)
806 CONTINUE
IF (JK-NB )815,814,815
814 DO 810 L=1,NBT
XRADI(L)=L
VNB=NB
VMAG2(L)= VMAG2(L)/VNB
VPHI2(L)= VPHI2(L)/VNB
810 AZ2(L)=AZ2(L)/VNB
ITITLE(1)=8HHD160/V
ITITLE(2)=8HEL600 /
ITITLE(3)=8HALT500/
ITITLE(4)=8HLEV/RT3
ITITLE(5)=8HD/S/T=1
ITITLE(6)=8H
ITITLE(7)=8H
ITITLE(8)=8H
ITITLE(9)=8HJOB0194
ITITLE(10)=8HDELANEY
ITITLE(11)=8H W. F.
ITITLE(12)=8HKALMON1
LABEL=4HAVRD
CALL PLOT (NUMB,XRAD1,VMAG2,0,0,LABEL,ITITLE,3.,0.,1,0,0,2,2, 9,6,
1 0, LAST)
LABEL=4HAVAZ
CALL PLOT (NUMB,XRAD1,VPHI2,0,0,LABEL,ITITLE,3.,0.,2,0,0,2,2, 9,6,
1 0, LAST)
LABEL=4HAVEL
CALL PLOT (NUMB,XRAD1,AZ2 ,0,0,LABEL,ITITLE,3.,0.,1,0,0,2,2, 9,6,
1 0, LAST)
815 PRINT 667
667 FORMAT(//,30X, 5HV MAG2 )
PRINT 668,(VMAG2(J), J=1,15)
668 FORMAT(/,1X,15F7.4)
PRINT 669

```

```
669 FORMAT(//,30X,5HVPHI2)
PRINT 668,(VPHI2(J),J=1,15)
PRINT 670
FORMAT(//,30X,3HAZ2)
PRINT 668,(AZ2(J),J=1,15)
RETURN
END
```

SUBROUTINE ABFILT

```

DIMENSION RANGE(10),TARGRNG( 10 ),PD(10),
1 ITITLE(12),HD(10),VEL(10),ALT(10),PITCH(10),XT( 10 ),YT( 10 ),
2 ZT(10),AZT(10),C(10,10),S(10),ELT(10),XRAD(10),YRAD(10),ZRAD(10),
3 ZLRAD(10),AZRAD(10),BOXMIN(10),BOXMAX(10),ELTEST(10),IAZ(10),
4 PRADRNG(10),PELRAD(10),PAZRAD(10),RADRNG(10),XT1(200),YT1(200),
5 XRAD1(200),YRAD1(200),INDEX(10),INDE(10),CONT(10),AT(10),PXRAD
6 (10),PYRAD(10),PZRAD(10),DOTX(10),DOTZ(10),DOTH(10),NR(10),HDFIX(
710),PITFIX(10),VELFIX(10),RG(10),AZG(10),ELG(10),R1(20),R2(20),
8 R3(20),R4(20),AZ1(20),AZ2(20),AZ3(20),AZ4(20)
COMMON RANGE,TARGRNG,PDET,PD,IR,RANDOM,NUNIF,DEV,
1 RANERR, RADRNG,HD,VEL,ALT,PITCH,XT,YT,NT,JA,I,TEST,
2 AZT , ELT, IAZ, BOXMAX,BOXMIN,ELTEST,TEST,AZRAD1,
3 ZRADRNG1,ELRAD1, AZRAD ,C,S,DOTT,ZT,PAZRAD,ELBEAMT
4,FLAG, NUMTRK, NUM1 ,NUM2,NUM3 ,AZBEAMT,RADDEG,ELRAD
5 , XRAD, YRAD, ZRAD,CONT,PRADRNG,PELRAD,XT1,YT1,XRAD1,YRAD1,INDEX,IN
6 DE,IAT,PXRAD,PYRAD,PZRAD,DOTX,DOTZ,DOTY,NR,HDFIX,PITFIX,VELFIX
7 ,ALPHA ,BETA,RG,AZG,ELG,R1,R2,R3,R4,AZ1,AZ2,AZ3,AZ4
DO 20 L=1,NUMTRK
DO 21 IC= 1,10
IF(C((IC,L)) 21,21,22
21 CONTINUE
N= 10
XRAD(N)= PXRAD(L)
YRAD(N)= PYRAD(L)
ZRAD(N)=PZRAD(L)
GO TO 23
22 N= IC
K=IC
XRAD(K)=(RADRNG(K)*COSF( ELRAD(K )/57.29577) ) *
1 SINF(AZRAD(K )/ 57.29577)
YRAD(K )=(RADRNG(K )*COSF(ELRAD(K )/57.29577) ) *
1 COSF(AZRAD(K )/ 57.29577)
ZRAD(K )=RADRNG(K )*SINF( ELRAD(K )/57.29577)
23 DO 1 JJ=1,3
1 IF(JJ-1) 3,2,3

```

```

2 OISE = XRAD(N)
  PRED = PXRAD(L)
  DOTT = DOTX(L)
  GO TO 6
3 IF (JJ-2) 4,5,4
5 OISE=PYRAD(N)
  PRED = PYRAD(L)
  DOTT = DOTY(L)
  GO TO 6
4 OISE = ZRAD(N)
  PRED = PZRAD(L)
  DOTT = DOTZ(L)
6 SMOOTH = PRED + ALPHA * ( OISE - PRED)
  DOTT= DOTT+ (BETA/T)*( OISE-PRED)
  PRED = SMOOTH + T*DOTT
  IF (JJ-1) 8,7,8
7 PXRAD(L) = PRED
  XRAD(N) = SMOOTH
  DOTX(L) = DOTT
  GO TO 1
8 IF (JJ-2) 9,96,9
96 PYRAD(L) = PRED
  YRAD(N) = SMOOTH
  DOTZ(L) = DOTT
  GO TO 1
9 PZRAD(L) = PRED
  ZRAD(N) = SMOOTH
  DOTZ(L) = DOTT
1 CONTINUE
537 IF (ABSF (PXRAD(L))-0.001) 533,534,534
533 PXRAD(L) = 0.001
534 AZ1=ATANF (PYRAD(L)/PXRAD(L)) *RADDEG
  IF (PXRAD(L)) 530,531,531
530 PAZRAD(L) = 270.0-AZ1
  GO TO 532
531 PAZRAD(L) = 90.0 -AZ1

```

```
532 PELRAD(L) = ATANF(PZRAD(L)/SQRTF((PXRAD(L))**2+  
1(PYRAD(L))**2)*RADDEG  
PRADRNG(L) = SQRTF((PXRAD(L))**2+(PYRAD(L))**2+(PZRAD(L))  
1**2)  
1 PRINT 201,PRADRNG(L),PAZRAD(L),PELRAD(L)  
201 FORMAT(/,10X,10HPRADRNG = ,F10.4,5X,9HPAZRAD = ,F10.4,5X,  
1 9HPELRAD = ,F10.4)  
20 CONTINUE  
515 RETURN  
END
```

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2. Library Naval Postgraduate School, Monterey, California	2
3. Commander, Naval Air Systems Command Hqs. Washington, D.C. 20360	1
4. Commander, Naval Ordnance Systems Command Hqs. Washington, D.C. 20360	1
5. Professor James S. Demetry Naval Postgraduate School, Monterey, California	2
6. LT William F. Delaney, 104 Spruce Street Watertown, Massachusetts 02172 c/o Mr. Robert Kelland	2

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
Naval Postgraduate School Monterey, California 93940		2b. GROUP
3. REPORT TITLE SIMULATION OF A TRACK-WHILE-SCAN RADAR		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Thesis		
5. AUTHOR(S) (Last name, first name, initial) Delaney, William F. Lieutenant United States Navy		
6. REPORT DATE June 1967	7a. TOTAL NO. OF PAGES 121	7b. NO. OF REFS 7
8a. CONTRACT OR GRANT NO.	8c. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned to report)	
d.		
10. AVAILABILITY/LIMITATION NOTICES [REDACTED]		
11. SUPPLEMENTARY NOTES		
12. SPONSORING MILITARY ACTIVITY Commander Naval Air Systems Command Hqs. Naval Ordnance System Command Hqs.		

13. ABSTRACT

A model of a TWS radar is developed that provides a realistic computer simulation for comparing various radar tracking methods.

Prediction accuracy of a simplified alpha-beta tracker is compared to that of an adaptive filter. In addition, the effect on radar tracking of a variable gate size correlation technique is investigated.

UNCLASSIFIED

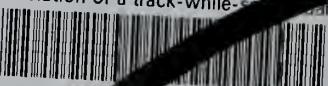
- Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Radar						
Filter						
Track-While-Scan						
Tracking						
Simulation						

[REDACTED]

thesD265

Simulation of a track-while-scan radar



3 270 001 02528 1

DUDLEY KNOX LIBRARY

DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01058549 0